



Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica

Universitat de Barcelona

**SHAREDUBT: APLICACIÓN ANDROID PARA
DEBATIR Y COMPARTIR DUDAS DENTRO DE
UNA COMUNIDAD DE ESTUDIANTES**

Gustavo A. Tenorio Ibarra

Director: Sergio Sayago

Realitzat a: Departament de

Matemàtiques i Informàtica

Barcelona, 27 de juliol de 2018

ÍNDICE

1. Introducción	6
1.1. Motivación	7
1.2. Objetivos	7
1.3. Novedades que aporta la aplicación foro respecto a aplicaciones y sistemas similares	8
2. Metodología	10
2.1. Planificación del proyecto	10
2.2. Desarrollo iterativo e incremental	11
2.2.1. Tecnología utilizada	12
2.2.2 Fases de desarrollo	13
2.2.3. Presupuesto	15
3. Análisis	16
3.1. Encuesta	16
3.2. Definición de la interfaz	19
3.2.1. Evaluación de interfaces similares	21
3.3. Prototipo conceptual	22
3.3.1. Evaluación del prototipo conceptual	23
3.4. Casos de usos	24
3.4.1. Registrarse	25
3.4.2. Iniciar sesión	25
3.4.3. Cerrar sesión	25
3.4.4. Consultar perfil	26
3.4.5. Crear nueva duda	26
3.4.6. Gestionar asignaturas	27
3.4.7. Subir imagen	27
3.4.8. Ver duda	28
3.4.9. Responder duda	28
3.4.10. Gestionar contenido	28
3.5. Modelo de dominio	29
4. Patrones	30
4.1. Diseño de interfaz gráfica	30
4.2. Diseño de programación	31
5. Desarrollo y resultados	33
5.1. Incremento 1: Inicio de sesión y registro	33

5.1.1. Requisitos funcionales y especificaciones	33
5.1.2. Diagrama de clase	34
5.1.3. Comportamiento	36
5.1.4. Pruebas del funcionamiento	38
5.2. Incremento 2: Foro básico.....	38
5.2.1. Requisitos funcionales y especificaciones	39
5.2.2. Diagrama de clase	40
5.2.3. Evaluación.....	42
5.2.4. Comportamiento	43
5.2.5. Pruebas del funcionamiento	44
5.3. Incremento 3: cursos y asignaturas.....	45
5.3.1. Requisitos funcionales y especificaciones	45
5.3.2. Diagrama de clase	46
5.3.3. Evaluación.....	48
5.3.4. Comportamiento	48
5.3.5. Pruebas del funcionamiento	50
5.4. Incremento 4: Perfil y respuesta avanzada.....	50
5.4.1. Requisitos funcionales y especificaciones	50
5.4.2. Diagrama de clase	51
5.4.3. Evaluación.....	52
5.4.4. Comportamiento	52
5.4.5. Pruebas del funcionamiento	53
6. Discusión	55
7. Conclusión.....	57
7.1. Perspectivas de futuro	57
8. Referencias.....	59
9. Anexos	60
9.1. Encuesta Google Forms	60

ABSTRACT

It is widely known that digital technologies are having a large impact on our daily lives, especially smartphones, which have become an essential technology for performing everyday activities. Hence, exists many apps designed to enrich and make our lives easier.

This project presents the conceptualization, design and development of ShareDoubt, an app for helping students of the degree of CS at University of Barcelona to share doubts and enrich their educational experience at university.

ShareDoubt tries to address one of the limitations of the Campus Virtual tool facilitated by the university: students do not use the forums despite being willing to share doubts online with their classmates. Therefore, the main objective is to create a students' community where, through the anonymity, all kind of doubts can be shared while doing the labs and preparing the exams. In this way, the participation of other users is encouraged, allowing the generation of different discussions from one doubt. These discussions can result in collective learning, enriching the knowledge of students.

For the creation of this community, an app for Android smartphones has been developed following an incremental work methodology, putting in practice what has been learned during the degree. The app has been evaluated with students in an iterative way. The results show that the app is useful and easy to use.

RESUM

Avui en dia la tecnologia té un gran impacte a la nostra vida quotidiana. Un exemple són els *smartphones*. Cada vegada són més persones que aprofiten les funcionalitats que aporta l'ús de tenir un telèfon intel·ligent a les seves mans i, d'una forma o una altra, s'ha fet imprescindible realitzar tota classe d'activitats des d'aquests dispositius gràcies a la seva facilitat d'ús. Per aquesta raó, existeix un considerable nombre d'aplicacions que pretenen oferir eines que simplifiquin i afavoreixin els diferents reptes a les quals ens enfrontem dia a dia.

L'ús de les aplicacions mòbils ha permès generar tecnologies d'informació i comunicació (TIC) que possibiliten compartir diferents classes de dades per així donar suport en molts aspectes de la nostra vida. Aquest projecte es basa en l'ús d'aquestes tecnologies dins de l'àmbit de l'educació, creant la plataforma ShareDoubt amb la finalitat d'ajudar als estudiants d'enginyeria informàtica de la Universitat de Barcelona a compartir dubtes.

ShareDoubt pretén cobrir la falta d'ús que els estudiants d'informàtica fan, en general, dels fòrums del Campus Virtual de la universitat, per la qual cosa l'objectiu general del treball és crear una comunitat d'estudiants que, mitjançant un pseudo-anonimat, puguin compartir tots els dubtes que els sorgeixi a l'hora d'estudiar mitjançant una app. D'aquesta forma, s'incentiva als usuaris a participar en diferents debats que inicien a partir d'un dubte creat per un alumne de la plataforma. Aquests debats poden derivar a un aprenentatge col·lectiu i enriquir el coneixement dels usuaris.

Per a la creació d'aquesta comunitat, s'ha desenvolupat una aplicació per als dispositius mòbils Android seguint la metodologia de treball incremental, posant en pràctica tots els coneixements adquirits al llarg del grau. A més, en el transcurs de la creació d'aquesta eina, s'ha dut a terme un estudi amb persones reals a partir de realitzar diferents avaluacions amb possibles futurs usuaris per així tenir en compte els seus comentaris i oferir un servei que s'assembli a les seves necessitats.

RESUMEN

Hoy en día la tecnología tiene un gran impacto en nuestra vida cotidiana, destacando el constante empleo de los smartphones. Cada vez son más personas que aprovechan las funcionalidades que aporta el uso de tener un teléfono inteligente en sus manos y, de una forma u otra, se ha hecho imprescindible realizar toda clase de actividades desde estos dispositivos gracias a su facilidad de uso. Por esta razón, existe un considerable número de aplicaciones que pretenden ofrecer herramientas que simplifiquen y favorezcan los diferentes retos a las que nos enfrentamos día a día.

El uso de las aplicaciones móviles ha permitido generar tecnologías de información y comunicación (TIC) que posibilitan compartir diferentes clases de datos para así dar soporte en muchos aspectos de nuestra vida. Este proyecto se basa en el uso de estas tecnologías dentro del ámbito de la educación, creando la plataforma ShareDoubt con el fin de ayudar a los estudiantes de ingeniería informática de la Universidad de Barcelona.

ShareDoubt pretende cubrir la falta de uso de los fóruns por parte de los estudiantes del Campus Virtual de la universidad, por lo que el objetivo general del trabajo es crear una comunidad de estudiantes que, mediante un pseudo-anonimato, puedan compartir todas las dudas que les surja a la hora de estudiar. De esta forma, se incentiva a los usuarios a participar en diferentes debates que inician a partir de una duda creada por un alumno de la plataforma. Estos debates pueden derivar a un aprendizaje colaborativo y enriquecer el conocimiento de los usuarios.

Para la creación de esta comunidad, se ha desarrollado una aplicación para los dispositivos móviles Android siguiendo la metodología de trabajo incremental, poniendo en práctica todos los conocimientos adquiridos a lo largo del grado. Además, en el transcurso de la creación de dicha herramienta, se ha llevado a cabo un estudio con personas reales a partir de realizar diferentes evaluaciones con posibles futuros usuarios para así tener en cuenta sus comentarios y ofrecer un servicio que se asemeje a sus necesidades.

1. INTRODUCCIÓN

ShareDoubt es una aplicación para móvil Android que tiene como principal objetivo cubrir las necesidades de aprendizaje, comunicación y relación de los estudiantes a la hora de compartir dudas de las diferentes asignaturas en la carrera de Ingeniería informática de la Universidad de Barcelona (UB). Estas asignaturas son todas las que ofrece el grado excepto las menciones, ya que estas están relacionadas con otras carreras y por ahora nos centraremos en ofrecer este servicio a las que hemos cursado y conocemos.

Esta herramienta se centra en poder ayudar a aquellos estudiantes que, durante la época de exámenes o durante las entregas, tengan dudas y no sepan cómo hallar respuestas. La finalidad es crear una comunidad de estudiantes donde se puedan resolver todo tipo de dudas a lo largo el grado y así conseguir que éste sea más llevadero.

El método de debatir y compartir cuestiones dentro de la comunidad de estudiantes que deseamos integrar es mediante el uso de foros. Un foro es un lugar físico o virtual que se emplea para reunirse e intercambiar ideas y experiencias sobre diversos temas. En el caso de nuestra aplicación, cada asignatura tendrá tantos foros como dudas se manifiesten en ella, en las que se podrá indagar mediante un pseudo-anonimato, una de las grandes ventajas que aportamos. Esta última característica en relación con el perfil del usuario, permite poder ayudar al compañero sin la necesidad de saber a quién ni quien ayuda, dado que no se reflejarán los datos personales de los estudiantes registrados.

Decidimos utilizar un foro virtual como base para la estructura de ShareDoubt ya que es una herramienta comunicativa que permite al usuario invertir el tiempo que desee en realizar dudas o en contestar la de otros compañeros de forma asíncrona, sin estar en una constante interacción con la aplicación y dejando las aportaciones recogidas permanentemente. De esta forma, evitaríamos el abuso de múltiples notificaciones que interfieran con el día a día del usuario, como sería el caso de aplicaciones de mensajería instantánea (Houston, H., Shewmaker, J. y Nguyen, J. 2017; Gikas J. 2013). Otro gran aporte de los foros es la posibilidad de participación de una forma reflexiva, frente a otras herramientas de comunicación, donde la inmediatez supone un obstáculo al raciocinio y el análisis (Mora, F. 2011).

Hay que destacar también, que cada uno de los foros nos puede derivar a un aprendizaje colaborativo, donde surja un debate que enriquezca el conocimiento de los participantes y con ello, el factor de la socialización.

La planificación del proyecto se ha basado en seguir el método visual de diagrama de Gantt. Por otro lado, la realización de un análisis de la tecnología utilizada en el ámbito escolar y la evaluación de aplicaciones similares a nuestro tema central, los foros, ha permitido establecer las bases de funcionalidad y diseño de la aplicación. Además, para extraer información sobre qué métodos llevan a cabo los estudiantes a la hora de estudiar, se ha tenido en consideración a estudiantes de diferentes facultades mediante una encuesta a través de Google Forms. Al ser una aplicación Android, su desarrollo se ha realizado con el IDE Android Studio y la parte del servidor apoyada con la plataforma Firebase¹. Este desarrollo ha seguido la metodología de

¹ <https://firebase.google.com>

trabajo incremental, donde se ha procurado seguir el prototipo diseñado y, a través de continuas entrevistas a lo largo del proceso, tener en cuenta la opinión de futuros usuarios. Durante la realización de dicha herramienta, se han aplicado diferentes conceptos aprendidos a lo largo del grado, como son: casos de uso, modelo de dominio, diagrama de clases y patrones de programación.

1.1. MOTIVACIÓN

Hoy en día, no cabe duda de que el móvil ha pasado de ser una herramienta para funciones básicas de comunicación a ser una extensión de nosotros mismos. Por esta razón, nuestra mayor motivación para este TFG es realizar una aplicación para estos dispositivos con el fin de aprender a desarrollar un producto que todos puedan obtener y utilizarlo desde sus manos. Por lo que, centrados en esto, el profesor Sergio Sayago me comentó que, hablando con estudiantes y algunos profesores de la facultad, veían que el uso del foro en el Campus virtual por parte de los estudiantes era mínimo y que serían más adecuadas otras vías de comunicación. De aquí surgió la idea de ShareDoubt.

Por otro lado, la realización de este proyecto ha sido todo un reto porque, aunque en la carrera nos dan la oportunidad de desarrollar aplicaciones Android, éstas son a escala local. El día que empezamos a realizar el TFG no habíamos visto cómo hacer un chat en tiempo real, ni cómo guardar archivos en la nube, entre otras cosas. Son cosas que siempre habíamos querido llegar a hacer y por esta razón, cuando supimos de todo lo que teníamos a disposición con la creación de la aplicación, la motivación fue aún mayor, pudiendo aprender cada una de las cosas que hemos llegado a implementar en el trayecto de este proyecto.

Por último, la idea de dar a cada uno de los estudiantes una herramienta que le sea útil para saber que no es el único que ha podido no entender algún tema de una asignatura y que de esta forma no se sienta solo, es más que suficiente para que hayamos desarrollado ShareDoubt.

1.2. OBJETIVOS

El objetivo general de este proyecto es diseñar e implementar una aplicación que se base en intercambiar ideas y debates sobre los contenidos de las asignaturas del grado en ingeniería informática de la UB. Aunque nos hemos centrado en este grado, porque es el que conocemos mejor, el diseño y desarrollo ha tenido en cuenta otros estudios que ofrece la UB. Más adelante expondremos que la problemática que aborda el TFG es importante también en diferentes grados.

Para lograr dicho objetivo y conseguir un correcto desarrollo de este, ha hecho falta efectuar diferentes sub-objetivos. Son los siguientes:

1. **Análisis previo de aplicaciones en el ámbito de la educación:** Buscar y leer información sobre el impacto que han tenido las aplicaciones móviles en la educación,

para así observar qué conlleva utilizar este tipo de herramientas para los estudiantes a la hora del estudio.

2. **Estudio de aplicaciones similares:** Estudiar las aplicaciones disponibles actualmente que pertenecen al mismo ámbito, foros, ayudará a localizar las desventajas y beneficios que tienen estas con el objetivo de conseguir hacer una aplicación que sea lo más robusta y funcional posible. Además, no sólo serán un estudio funcional de aplicaciones similares, también se extraerá información que ayude al diseño de la aplicación.
3. **Realizar una evaluación del diseño:** Entender las diferentes necesidades y gustos de quién será el usuario final de la aplicación es fundamental para poder generar un buen prototipo que encamine hacia la aplicación definitiva.
4. **Implementar el código:** Desarrollar la aplicación, probando las funcionalidades implementadas y entregando un producto lo más sólido y estable posible. Este es sin duda el sub-objetivo más difícil de hacer y el más motivador.

1.3. NOVEDADES QUE APORTA LA APLICACIÓN FORO RESPECTO A APLICACIONES Y SISTEMAS SIMILARES

Actualmente se usa la aplicación WhatsApp, o similares, para crear grupos en los cuales comentar sobre diferentes temas, esto es perfectamente aceptable, pero siempre se desea poder dividir el ocio y el trabajo. A pesar de que también existan aplicaciones destinadas a la comunicación entre compañeros de trabajo, como puede ser Slack², este tipo de herramientas están destinadas a mensajes instantáneos, donde se hacen preguntas simples y varias respuestas rápidas en diferentes líneas de texto. En cambio, en los foros como ShareDoubt, se puede postear una pregunta laboriosa y esperar a que otros usuarios puedan meditar su respuesta y contestar a ésta en un único mensaje. Además, las discusiones que no están posteadas en foros suelen entrelazarse con otro tipo de conversaciones al mismo tiempo que hacen difícil de seguir un el hilo de la conversación. Por esta razón, ShareDoubt, como la mayoría de los foros, consta de un moderador para poder tener un control de la información y así evitar que el tema principal de la discusión se desvíe.

ShareDoubt, al ser una aplicación basada en foros, permite guardar las discusiones que hayan de las diferentes dudas y así los futuros usuarios que puedan tener esa misma duda podrán ver toda la información compartida o simplemente dar su opinión al respecto. Esto con aplicaciones de mensajes instantáneos también es posible hacerlo, pero de una forma más incómoda, teniendo que desplazarse por toda la conversación hasta llegar al foco deseado.

Esta aplicación también te da la posibilidad de crear una comunidad de confianza donde anónimamente, bajo el seudónimo que decidas, poder preguntar tus dudas y aportar tus consejos en los diferentes foros con la gente de tu ámbito. Dicha confianza viene dada a que el usuario tiene una privacidad porque su respuesta no está relacionada a alguien físico. Es decir, es tu usuario de la comunidad y sólo existes dentro de ésta.

² <https://slack.com/>

Existen diferentes plataformas para compartir apuntes y foros especializados en algún tema específico como pueden ser, coches, juegos, comida, entre otros, pero no hemos encontrado ninguno aplicado a la carrera de ingeniería informática de nuestra universidad. Además, la mayoría de foros pueden ser algo más generales y en ShareDoubt se pretende que el usuario pueda hablar con gente que esté o haya estado en su misma situación. No obstante, esto da posibilidad a poder copiar ejercicios de compañeros, pero el fin de esta aplicación es obtener ayuda, por lo que el uso que le den sus usuarios con el material obtenido ya estaría fuera del control del moderador, el cual seríamos nosotros como los creadores.

Por último, a diferencia del foro del Campus Virtual, esta plataforma permite al usuario expresar dudas de un tema que no haya entendido sin pasar esa posible vergüenza por preguntar algo que el profesor ha explicado anteriormente en clase. Por esta razón, el foro que facilita la propia universidad suele ser visitado en gran parte sólo por los profesores. Esto último se puede contrastar en el apartado 3.1. Encuesta.

2. METODOLOGÍA

2.1. PLANIFICACIÓN DEL PROYECTO

Para plantear el desarrollo de este trabajo realizamos un diagrama de Gantt, tanto para la planificación inicial (Figura 1) como para la real (Figura 2). Este tipo de diagrama es una herramienta gráfica que tiene como objetivo exponer el tiempo de dedicación para diferentes tareas a lo largo de un tiempo total determinado. Este método nos permite interpretar el proyecto de una forma fácil y poder llevar un control de la planificación.

Las tareas comienzan con una investigación sobre cómo afectan las aplicaciones en los estudiantes para así tener una idea de qué nos aporta la creación de ShareDoubt en nuestra facultad. Después de este primer análisis, pasamos a escribir una parte de la memoria, introduciendo el trabajo que queremos hacer, qué objetivos nos marcamos para ello, cómo y con qué nos planteamos realizarlo y, por último, un estudio con usuarios reales para así obtener un prototipo de la aplicación y empezar su desarrollo. Para el desarrollo nos hemos basado en el modelo iterativo incremental y, como consecuencia, hemos creado incrementos, los cuales se detallan en el apartado 2.2.2 Fases de desarrollo. Hemos escogido este tipo de metodología de trabajo con la idea de ir adquiriendo conocimiento sobre la materia e ir evolucionando y superándonos con cada entrega de los diferentes incrementos. Una vez tengamos la aplicación terminada, finalizaremos el escrito de la memoria con una discusión comparando los objetivos y resultados, un análisis de costes, concluiremos el escrito y expondremos todas las referencias utilizadas para la realización del trabajo. Por último, nos prepararemos la defensa del trabajo ante el tribunal.

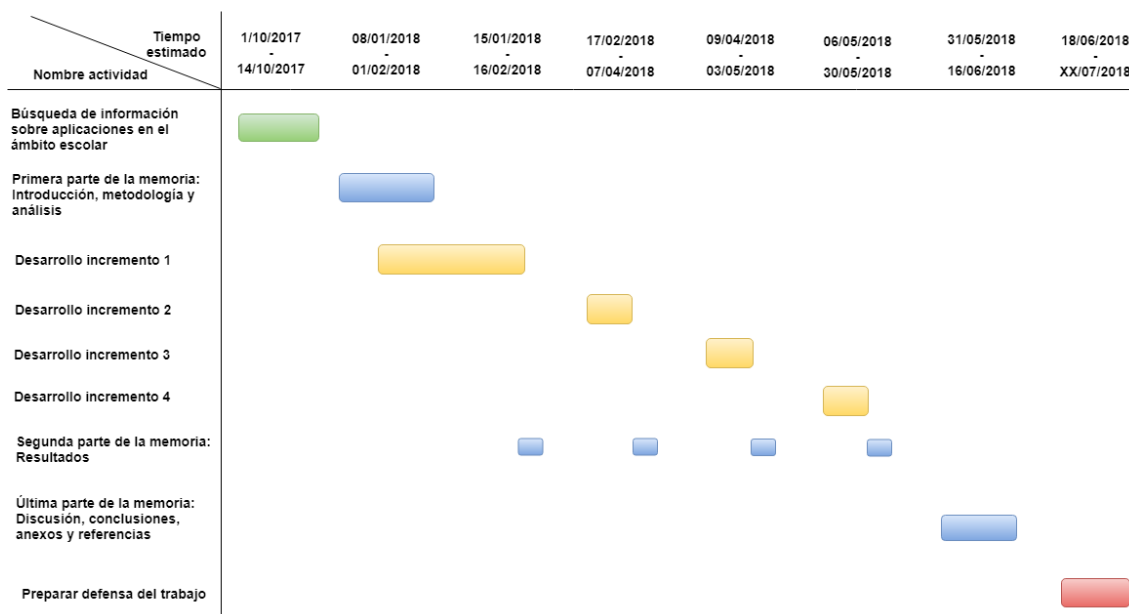


Figura 1. Diagrama de Gantt, planificación inicial

Tras observar la planificación inicial, vamos a exponer el diagrama de Gantt con la planificación real en la que finalmente se ha desarrollado el proyecto. La realización de una aplicación Android de estas características sin ningún tipo de experiencia previa hizo que el período de tiempo inicialmente destinado no fuera lo suficientemente extenso. Es por esto por lo que hemos observado grandes diferencias entre la planificación inicial y el tiempo real invertido, sobre todo en cuanto a finalizar los diferentes incrementos. Esto nos ha ayudado a comprender mejor el peso de cada apartado del trabajo a realizar, asegurando una mejor planificación en futuros proyectos.

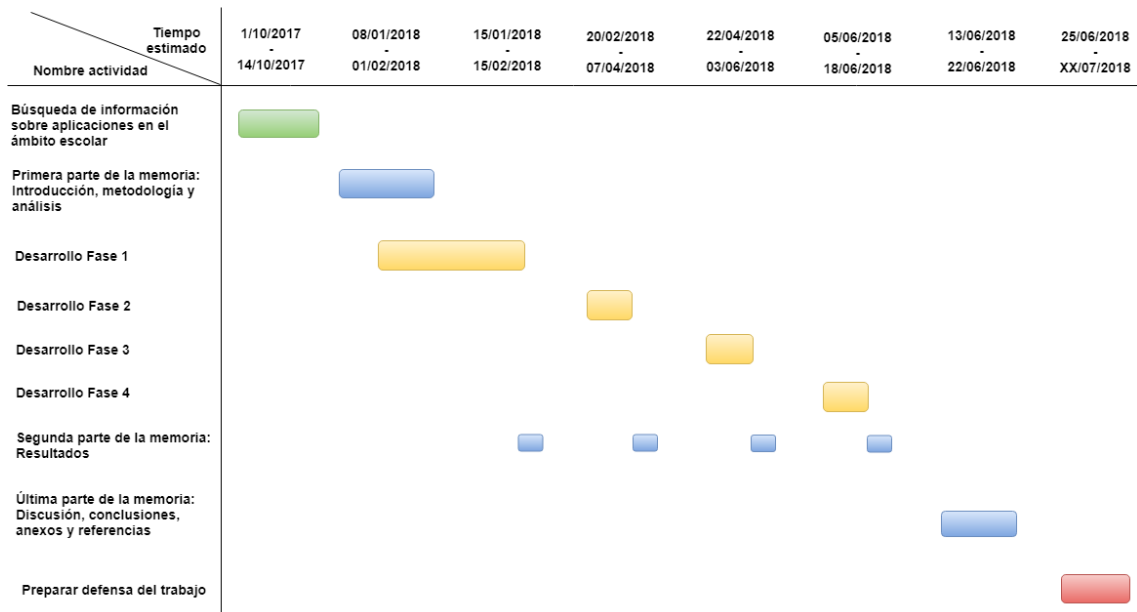


Figura 2. Diagrama de Gantt, planificación real

2.2. DESARROLLO ITERATIVO E INCREMENTAL

En cuanto a la metodología de desarrollo de software, nos hemos basado en el modelo iterativo incremental (Sommerville, I. 2011). Una metodología de programación basada en realizar la aplicación en diferentes fases de desarrollo (Figura 3) de manera que haya: una inicialización con los requisitos principales que se buscan en el proyecto; períodos de incremento a lo largo del desarrollo; lista de control conforme se vaya realizando cada incremento para llevar un dominio de ésta.

Esta forma de trabajo nos permite obtener un *feedback* con nosotros mismos cada vez que realizamos una entrega del incremento. Esto permite corregir o mejorar cualquier aspecto que se crea más oportuno y verificar que todo se está haciendo según lo previsto, siempre con el fin de obtener un software mejor que el anterior al final de cada uno de los incrementos.

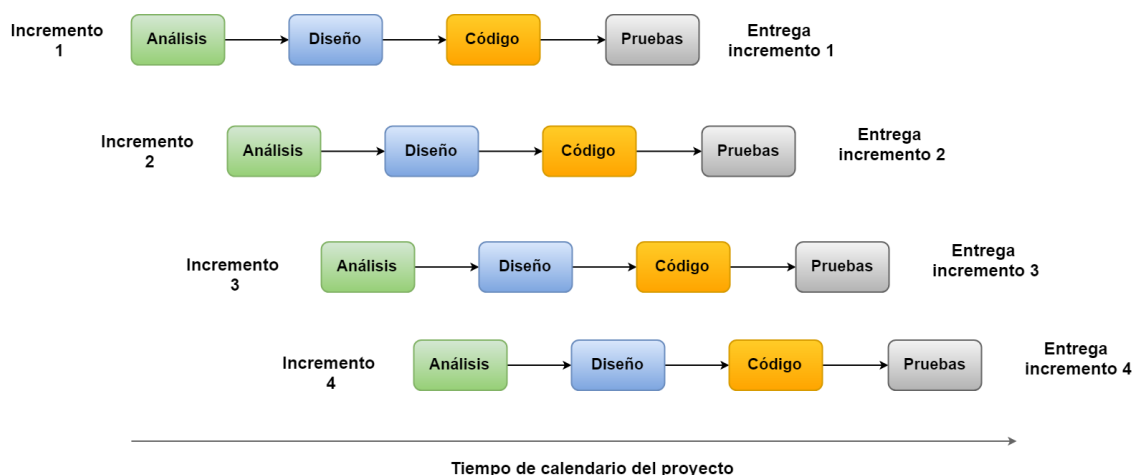


Figura 3. Desarrollo iterativo e incremental y sus fases

2.2.1. TECNOLOGÍA UTILIZADA

Antes de empezar a detallar la tecnología usada para el trabajo, cabe aclarar que hemos decidido realizar el desarrollo en Android porque es el sistema operativo más popular utilizado en los móviles, con un 76.53% en el último año (Statcounter³, mayo 2018). Además, al poder programar con el lenguaje Java, que es el que más hemos utilizado en el grado, nos es más factible llevar a cabo la creación de ShareDoubt.

Al ser una aplicación destinada a móviles con este sistema operativo, se ha utilizado el IDE Android Studio para la creación de ésta. Este entorno, desarrollado por Google, utiliza el lenguaje de programación Java juntamente con XML, para así implementar todas las funcionalidades lógicas y el diseño de la aplicación, respectivamente, con mayor comodidad. Hemos utilizado este IDE ya que es el oficial para programar Android y el que más soporte tiene. Además, al haber hecho uso de él en la carrera, ya estábamos familiarizados.

Uno de los soportes que hemos utilizado para realizar la aplicación es la plataforma Firebase, también ofrecida por Google, que ofrece una API para el desarrollo en la nube donde se permite tener datos sincronizándose en tiempo real a través de múltiples dispositivos, una de las tareas que pueden llevar más tiempo a la hora de programar. Además, tiene un propio sistema de comprobación de correos registrados, almacenamiento de archivos de los usuarios y diferentes funciones que facilitan la creación del software. Cabe comentar que la base de datos a tiempo real que utiliza Firebase es NoSQL, a diferencia de la base de datos relacional aprendida en el grado.

Para realizar los *mockups* de la interfaz de la aplicación, hemos utilizado Balsamiq. Esta herramienta nos permite realizar el prototipo del proyecto en forma de bocetos para así tener una imagen visual de cómo será nuestra interfaz gráfica. Además, hemos empleado Invision para así integrar este prototipo en una aplicación y poder llevar a cabo una evaluación con usuarios reales.

³ <http://gs.statcounter.com/os-market-share/mobile/worldwide/>

Para el desarrollo de los diagramas realizados en esta memoria, nos ha sido muy útil la herramienta proporcionada en la página web draw.io⁴. Se pueden crear diagramas UML de una forma fácil y rápida.

Por último, para tener un control de versiones y mantener el código almacenado en un repositorio, hemos utilizado GitHub. Esta herramienta nos facilita tener a salvo nuestro proyecto y poder tener un completo acceso a él desde cualquier ordenador.



Figura 4: Tecnología utilizada

2.2.2 FASES DE DESARROLLO

Para poder llevar a cabo la aplicación, nos centramos en aplicar diferentes etapas para el desarrollo de *software*, para así estructurar la aplicación a desarrollar y enfocar su construcción de la mejor forma posible. Las fases son las siguientes:

1. **Análisis de los requerimientos y especificaciones:** Describir el comportamiento que tendrá el programa.
2. **Diseño de diagrama de clases:** Describir cómo se relacionan los elementos del sistema a desarrollar para así seguir una arquitectura base del sistema.
3. **Diseño del prototipo:** Implementar en XML un diseño avanzado del prototipo extraído con el estudio a usuarios reales.
4. **Programación:** Llevar a cabo toda la lógica de lo diseñado.
5. **Pruebas:** Realizar un conjunto de pruebas para validar el funcionamiento de aplicación.
6. **Documentación:** Escribir los resultados obtenidos en la memoria.

⁴ <https://www.draw.io/>

Todas estas fases comentadas forman lo que denominamos un incremento. Para realizar el desarrollo de la aplicación de una forma más particionada y priorizada, hemos creado cuatro incrementos:

- **Incremento 1:** Inicio de sesión y registro. Esto conlleva a crear toda la gestión para almacenar todos los datos de los usuarios y utilizar estos para acceder a la aplicación siempre y cuando estén en la base de datos.
- **Incremento 2:** Foro simple. Generar una aplicación base que permita crear diferentes foros en los cuales se puedan establecer una discusión por alguna duda y responder a ésta.
- **Incremento 3:** Cursos y asignaturas. Después de tener una base con un conjunto de foros, añadir la organización de dichos foros por cursos y asignaturas. De esta forma habrá una gestión de los foros.
- **Incremento 4:** Perfil y respuesta avanzada. Dar una sección al usuario donde tener su área personal y la posibilidad de adjuntar una imagen a su respuesta.

A continuación, expondremos las causas de haber decidido poner una prioridad y un orden a cada uno de estos incrementos:

Para comenzar, hemos puesto como primer incremento el desarrollo de las pantallas de iniciar sesión y registro ya que necesitamos tener un sistema de ingreso a la aplicación e irnos familiarizando con el entorno de desarrollo y con nuestro servidor, Firebase. De esta forma tendremos una base de datos con usuarios y más conocimiento sobre el tema.

Como segundo incremento, hemos agregado la posibilidad de crear dudas (sin tener ninguna relación con alguna asignatura) y poder contestar a estas de forma rápida con el propósito de desarrollar primero el sistema de foros y así obtener una base con la cual poder trabajar. En otros términos, tener como mínimo una aplicación que te permite realizar en gran parte en lo que se basa ShareDoubt, los foros. En este incremento ya es necesario crear la barra de navegabilidad para navegar entre la pantalla de creación y visualización de los foros.

Ya con una aplicación funcional, nos encontramos con el tercer incremento. En este punto ya podemos pasar a desarrollar la parte de cursos, cada uno con sus respectivas asignaturas y éstas con las dudas compartidas por los usuarios de la aplicación. Es el incremento que le añade algo de individualidad a la aplicación por el hecho de tener una pantalla personal con las asignaturas de interés por parte del usuario. En este punto podemos decir que tenemos el proyecto consolidado y podemos pasar al cuarto incremento con tal de llegar al desenlace.

Por último, el cuarto incremento se basa en crear un apartado de perfil para el usuario, donde pueda modificar su imagen de perfil, ver las dudas que él ha puesto, la característica de respuesta avanzada y asegurar que toda la aplicación se encuentra en correcto funcionamiento.

2.2.3. PRESUPUESTO

La realización de este proyecto ha tenido un total de 180 días, lo que supone haber invertido un total de 720 horas, ya que hemos trabajado 4 horas diarias.

Por experiencia propia, sabemos que un programador junior cobra sobre 8 euros/hora, y en nuestro caso, todas las funcionalidades que se describen en la Tabla 1 han sido realizadas por éste, además del desarrollo de la aplicación. Como se puede observar en la Figura 5, la mayor parte del proceso en este proyecto ha sido dedicada a la programación de la aplicación. Las horas dedicadas se desglosan a continuación:

FUNCIONALIDAD	HORAS DEDICADAS (H)	PRECIO POR HORA (€/H)	PRECIO TOTAL (€)
ANÁLISIS	15	8	120
DESARROLLO	641	8	5128
EVALUACIÓN	4	8	32
DOCUMENTACIÓN	60	8	480
	720		5760

Tabla 1. Desglose de funcionalidades por horas y precio de cada una del proyecto

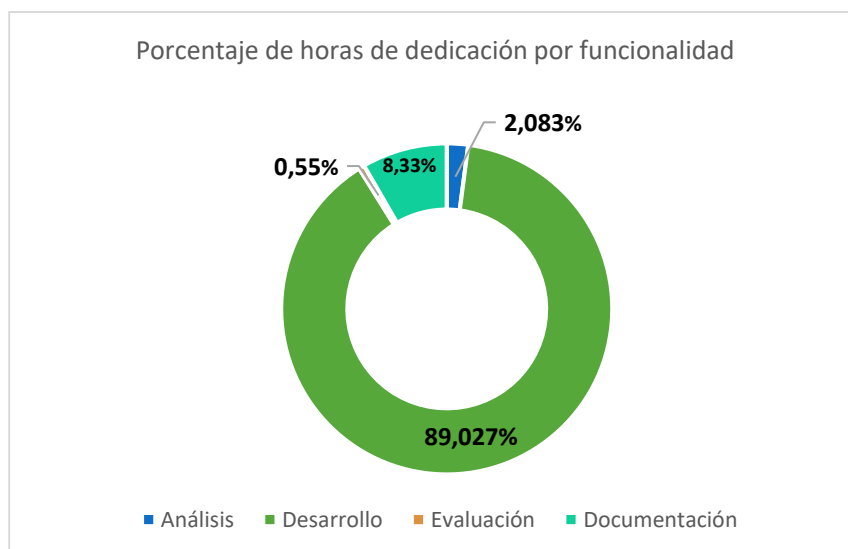


Figura 5. Diagrama del porcentaje de las horas dedicadas a cada funcionalidad del proyecto

3. ANÁLISIS

3.1. ENCUESTA

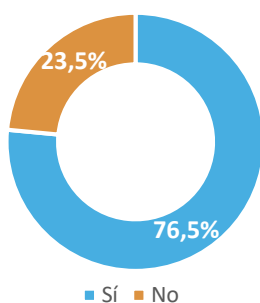
En este apartado buscamos tener la opinión de personas reales que nos ayuden a comprender cómo resuelven sus dudas y con ello, comprobar que la creación de ShareDoubt puede ser una herramienta que les de soporte a la hora de estudiar. No hemos querido centrar el estudio sólo sobre los alumnos que utilizarán la aplicación principalmente, sino que hemos intentado obtener una visión global sobre el uso de una aplicación para compartir apuntes entre los estudiantes de algunos grados de la Universidad de Barcelona. Además, queremos saber qué se está haciendo actualmente cuando un estudiante tiene una duda que no puede resolver por sí mismo.

El procedimiento para llegar a los diferentes usuarios fue mediante una encuesta generada y compartida mediante Google Forms, de esta forma el usuario puede contestar bajo ningún tipo de presión y en el momento que crea más oportuno. Es un método asequible para todos y permite llegar a muchas personas, asimismo de poder recopilar la información de una forma fácil y sencilla. Las preguntas realizadas tienen como objetivo principal saber los métodos más utilizados para estudiar entre los estudiantes de la Universidad de Barcelona y entender por qué la herramienta del campus virtual, facilitada por la propia universidad, no se emplea para compartir dudas entre estudiantes y profesores. Además, nos ayudará a entender la opinión de los estudiantes sobre la creación de una aplicación que permita compartir dudas y apuntes de manera anónima.

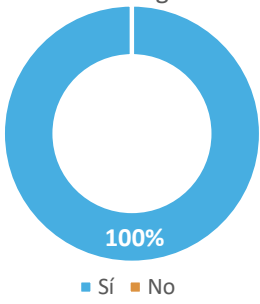
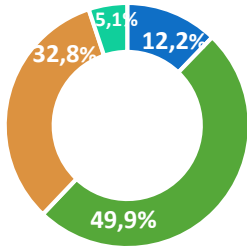
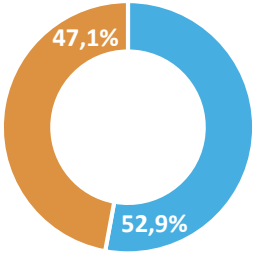
Después de poco más de tres semanas de estudio y un total de 103 encuestados, se ha podido obtener los resultados que se muestran en la Tabla 2. Datos de la encuesta y los perfiles de los que han contestado la encuesta son los siguientes:

- 13 estudiantes de física, último año.
- 18 estudiantes de biomedicina, último año.
- 21 estudiantes de enfermería, último año.
- 12 estudiantes ciencias ambientales, último año.
- 14 estudiantes de filología.
- 25 estudiantes de matemáticas e ingeniería informática, último y tercer año.

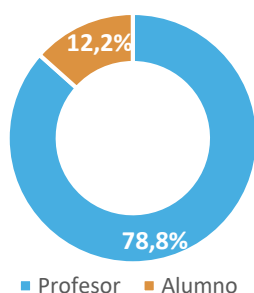
¿Eres de los que tienen vergüenza a preguntar en clase?



Podemos observar que la mayoría de los encuestados tienen vergüenza de preguntar dudas en clase y realmente es un dato para tener en cuenta ya que la mayoría son personas de último año y deberían de tener suficiente soltura para preguntar en clase.

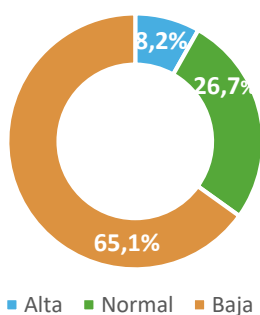
<p>¿Te gustaría poder tener la opinión de compañeros que hayan hecho o estén cursando la asignatura?</p>  <p>100%</p> <p>■ Sí ■ No</p>	<p>Vemos que el 100% apoya la idea de tener la apreciación de compañeros.</p>
<p>¿Qué herramientas sueles utilizar para compartir tus dudas?</p>  <p>49,9%</p> <p>■ Campus Virtual ■ Whatsapp ■ Facebook ■ Otros</p>	<p>Pese a que el Campus Virtual tenga un fórum destinado a cada asignatura, WhatsApp es la más utilizada por los estudiantes para compartir sus dudas y Facebook, con sus grupos de asignaturas, le sigue de cerca.</p>
<p>¿Te importaría compartir apuntes si es de forma recíproca?</p>  <p>52,9%</p> <p>■ Sí ■ No</p>	<p>Es la pregunta que tiene más discrepancia por parte de los diferentes encuestados. La mayoría de las respuestas negativas provienen por parte de carreras relacionadas con ciencias de la salud.</p>

¿Sueles compartir tu duda con un compañero o un profesor?



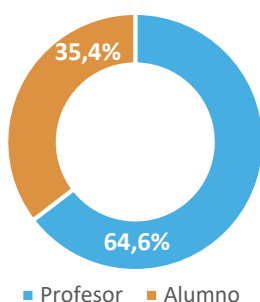
Hay que destacar de forma positiva que los encuestados aprovechan la función del profesor de resolver las dudas que tiene el alumno. Por lo que sería importante que los profesores pudieran formar parte de la aplicación si fuera necesario.

¿Con qué frecuencia sueles utilizar el foro del Campus Virtual?



Observamos que la frecuencia en el Campus Virtual es más bien baja. Este dato corrobora que hace falta alguna herramienta o una mejora en el foro del Campus Virtual para así incentivar al alumno a que lo utilice de forma más habitual.

¿Hacia quién crees que está destinado el foro del Campus Virtual?



Parece ser que los encuestados ven el Campus Virtual más enfocado para los profesores que para ellos. Seguramente sea porque el profesor suele usarlo más para compartir información sobre su asignatura.

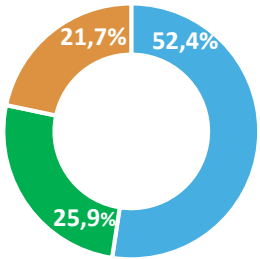
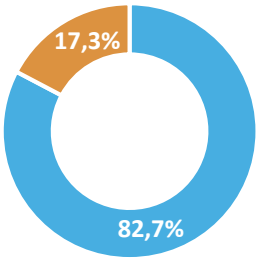
<p>¿A qué crees que es debido el poco uso del foro del Campus Virtual?</p>  <p>■ Timidez ■ Pregunta a profesor ■ Poco activo</p>	<p>Poco más del 50% cree que el Campus Virtual es poco usado por la timidez de la gente. Esto ocurre porque queda rastro de quién lo utiliza. También hay que decir que ¼ parte prefiere preguntar presencialmente al profesor.</p>
<p>¿Utilizarías una app que te permita compartir tus dudas de forma anónima dentro comunidad de alumnos?</p>  <p>■ Sí ■ No</p>	<p>Parece ser que no a todos les parece una buena idea tener una herramienta como ShareDoubt, pero por suerte para nosotros hay más abundancia de síes.</p>

Tabla 2. Datos de la encuesta

3.2. DEFINICIÓN DE LA INTERFAZ

El diseño de interfaz es el medio con el que el usuario interactúa con la aplicación y, por lo tanto, es importante tener bien definido cómo va a ser dicho diseño para la usabilidad de la plataforma. En este apartado, nuestro objetivo es realizar una interfaz con elementos claros y concisos, que den al usuario una agradable sensación mientras navega por la aplicación. Para ello, nos basamos en diseños minimalistas, destacando los pequeños detalles, sin sobrecargas visuales y basándonos en ofrecer un contenido fundamentado en la experiencia del usuario para que así le sea familiar su uso. En definitiva, buscamos conservar los principios de la ergonomía de Android para maximizar la eficiencia y la calidad de la interfaz.

Para realizar la interfaz de ShareDoubt, nos centramos en analizar diferentes diseños de aplicaciones que proporcionan una navegabilidad fácil y un acceso ágil a la información, con el fin de evitar la frustración del usuario a la hora de utilizar la aplicación. Las aplicaciones estudiadas y que se utilizaron para dar forma al diseño deseado de nuestra aplicación, son las siguientes:

1. **Instagram:** Esta famosa red social (Figura 6A) tiene una interfaz minimalista basada en los colores blanco y negro. Por lo que no abusa de colores y se centra en lo que queremos nosotros en nuestro diseño. Además, cuenta con una cómoda barra para el usuario, de modo que pensamos que es una forma fácil y distinguida para navegar entre las ventanas principales de la aplicación.
2. **Biotechnology Forums⁵:** Es una comunidad creada para discutir diferentes temas sobre el campo de la biotecnología (Figura 6B). Al basarse también en foros y en un tema en concreto como queremos, nos aportó diferentes ideas de cómo realizar el diseño y el funcionamiento que debería tener la aplicación. Lo que más nos interesó fue la distribución que tiene de mostrar los diferentes foros en su *timeline*, es una forma sencilla que contiene la información precisa de cada foro.
3. **Tapatalk⁶:** Sin duda, si hablamos de foros, la mayor plataforma para verlos y gestionarlos es Tapatalk (Figura 6C). Quisimos observar el diseño de la mayor comunidad de gestión de foros que existe hoy en día.

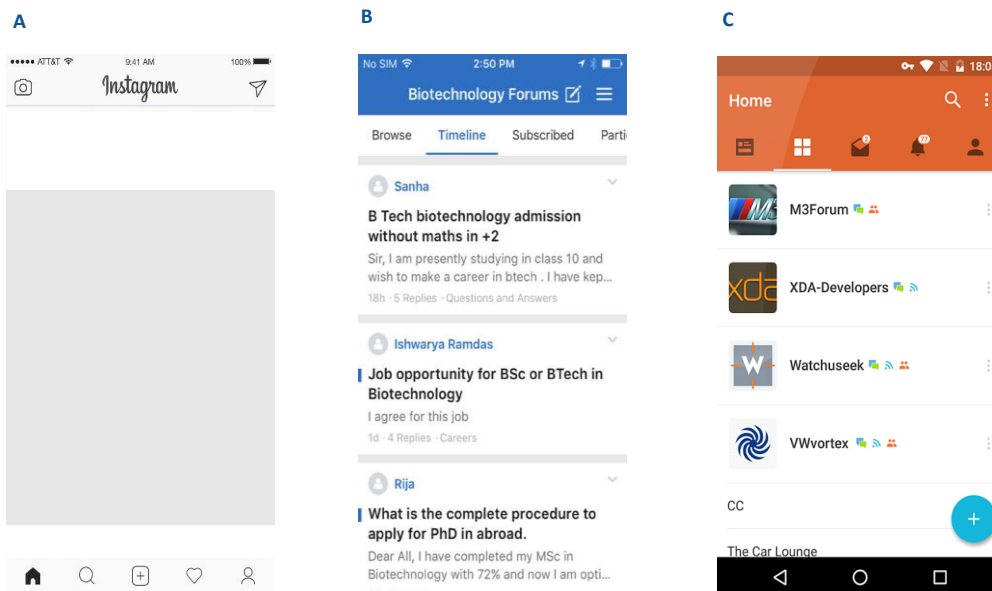


Figura 6: Aplicaciones estudiadas

⁵ Disponible en PlayStore:

<https://play.google.com/store/apps/details?id=com.tapatalk.biotechnologyforumscom>

⁶ Disponible en PlayStore:

<https://play.google.com/store/apps/details?id=com.tapatalk.biotechnologyforumscom>

Para poder extraer toda la información que requiere el proceso de definir la interfaz de una aplicación, realizamos una indagación contextual con algunos de los posibles usuarios futuros de ShareDoubt, tal y como aprendimos en la asignatura de Factores Humanos. Este estudio lo llevamos a cabo con dos de estas tres aplicaciones ya que con el diseño de Instagram ya estábamos familiarizados y teníamos claro que nos gustaba su barra de navegación, por lo que decidimos centrarnos en realizar una indagación en las otras dos que no las conocíamos y queríamos saber más sobre su diseño con foros.

3.2.1. EVALUACIÓN DE INTERFACES SIMILARES

Es fundamental que el usuario forme parte del proceso de diseño para así conseguir diversas consideraciones y comprender el uso que le dará a la aplicación éste. Es por esto por lo que llevamos a cabo una observación directa con cada uno de ellos, por separado, para ver la interacción y opinión sobre las interfaces similares. La observación se desarrolló en un espacio aislado para aumentar la concentración del observador y el observado.

Los observados fueron en total cinco usuarios, en un rango de edad de 22 a 24 años, estudiantes de ingeniería informática, con una gran experiencia con aplicaciones. Estos usuarios aportaron datos cualitativos de primera mano, por el hecho de ver los procedimientos de las tareas que les dimos a cumplir con estas dos aplicaciones. Las tareas fueron las siguientes:

- Publica un nuevo foro
- Comentar en un foro
- Buscar tema de *Bioinformatics*
- Visualizar mi perfil

Mientras realizaban estas tareas, fuimos redactando todas sus reacciones, comportamientos y datos que creíamos importantes. El tiempo que nos llevó hacer esta indagación fue aproximadamente de 10 minutos por aplicación, con cada usuario, ya que fueron tareas sencillas y con gente habituada al uso de esta tecnología.

Una vez analizamos los datos extraídos en las diferentes observaciones, pudimos observar que la aplicación debía tener ciertos aspectos que habían gustado a los usuarios y de otros que nos propusieron como usuarios avanzados en diferentes aplicaciones móviles. Los aspectos que más destacaron son los siguientes:

- **Respuesta rápida y completa:** Biotechnology Forums da la posibilidad de responder en un foro con una respuesta rápida, escribiendo un texto, o una respuesta completa, añadiéndole un archivo a tu respuesta.
- **Crear una duda:** En la misma aplicación que en el punto anterior, los usuarios han visto que les parece bien tener en la barra de navegación la opción de crear una nueva duda en vez de un botón flotante. También destacan que la pantalla de creación de un foro

de esta aplicación es muy usable y sólo le añadirían la posibilidad de seleccionar múltiples imágenes en el momento de adjuntar.

- **Ver consulta:** El formato que tiene Tapatalk a la hora de mostrar la consulta y las respuestas no les convence porque tanto la consulta como la respuesta tienen un mismo formato. Quieren que haya diferencia entre estas dos y no sólo se diferencie la consulta por ser el primer “comentario”.
- **Número de foros:** Tapatalk muestra de forma detallada cuántos foros hay sobre un tema.

Después de todas estas observaciones y sugerencias, pudimos dar comienzo a la elaboración de un prototipo conceptual, siempre pensando en los requerimientos que el sistema de la aplicación debe tener para ser aceptada por los usuarios. Estos requerimientos pueden ser funcionales y no funcionales (Arias Chaves, M. 2005). A continuación, se exponen los dos tipos y detallamos los requerimientos no funcionales utilizados para todos los incrementos, ya que estos son compartidos, al contrario de los requerimientos funcionales, que se encuentran detallados en cada uno de los incrementos en el apartado de 5. Desarrollo Y resultados.

- **Requerimientos funcionales:** Son los que especifican las funciones que el sistema podrá efectuar. Es importante que se describa el qué y no el cómo se debe hacer estas funciones. Estos requerimientos, al tiempo que avanza el proyecto de software, se convierten en los algoritmos que forman parte del desarrollo del sistema.
- **Requerimientos no funcionales:** Tienen que ver con características que de una u otra forma puedan limitar el sistema, como, por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad, mantenimiento, entre otros. Nuestra aplicación debe contener los siguientes requisitos no funcionales:
 1. El dispositivo móvil debe tener conexión a internet para así poder interactuar con el contenido de la aplicación.
 2. El dispositivo móvil debe tener una memoria RAM de 512 MB o superior y un espacio de almacenamiento mínimo de 15 MB.
 3. El dispositivo móvil debe tener una versión Android igual a 4.0.3 o superior.

3.3. PROTOTIPO CONCEPTUAL

El prototipo conceptual se basa en representar, de forma limitada, el diseño de un producto que permite al creador experimentar el uso, probarlo en situaciones reales y explorar su comportamiento. Está diseñado para satisfacer requerimientos funcionales y es muy útil e importante realizar un prototipo para así obtener un *feedback* de los usuarios antes del desarrollo del *software*.

En este prototipo pretendimos que el usuario tuviera un primer *look and feel* con las pantallas principales de la aplicación y para ello hicimos diferentes *mockups* con Balsamiq (Figura 7Figura 7). Una vez acabados, insertamos estos *mockups* en la plataforma Invision⁷, que permite navegar entre pantallas y así poder llevarlos a las manos de futuros usuarios mediante su aplicación.

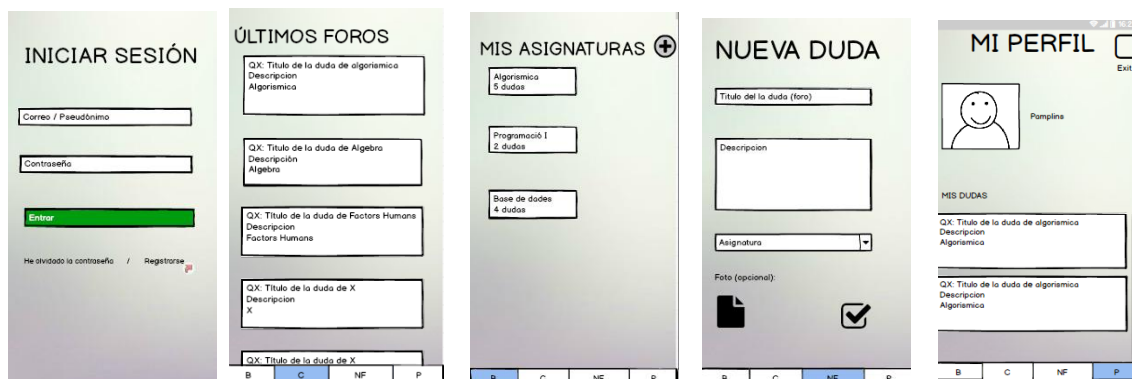


Figura 7: Pantallas principales del prototipo conceptual

3.3.1. EVALUACIÓN DEL PROTOTIPO CONCEPTUAL

Dado que la evaluación con aplicaciones similares fue muy útil, realizamos otra evaluación con el prototipo conceptual con los mismos usuarios, misma metodología de trabajo y un entorno similar que con la de interfaces similares. Por lo que a cada uno de ellos les pedimos realizar tres tareas con este prototipo para que así vieran las diferentes funcionalidades que tiene la aplicación y nosotros observar cómo se manejaban por ésta. Las tareas fueron las siguientes:

- Observa la primera duda de la asignatura “Algorísmica” utilizando la pantalla “Mis asignaturas”.
- Crea una nueva duda correspondiente a la asignatura “Algorísmica”.
- Añade la asignatura Álgebra a “Mis asignaturas”.

Finalmente, pudimos obtener diferentes comentarios que nos ayudaron a concretar los aspectos importantes de la navegabilidad en la aplicación y pudimos elaborar el diseño final de la aplicación tal y como se puede observar en el apartado de 5. Desarrollo Y resultados. Los comentarios que predominaron en la evaluación fueron los siguientes:

- **Imagen de usuario:** La posibilidad de poder añadir una imagen de usuario a la hora del registro. Al principio nuestra idea era poner una silueta de un usuario y que fuera común para todos, pero hemos visto que desean poner la imagen que ellos deseen y poder modificarla si es preciso en la pantalla de “Mi perfil”.

⁷ Proyecto del prototipo en Invision: <https://projects.invisionapp.com/share/YWF9R7ZHC#/screens>

- **Añadir nueva duda:** La posición donde está el botón para añadir una nueva duda no les ha gustado. Prefieren que dicho botón esté arriba a la derecha. Esto también afecta al botón de añadir una nueva asignatura.
- **Votar respuesta y ordenar:** Añadir la funcionalidad de poder votar las respuestas, tanto de forma positiva como negativa y con ello poner la que tenga más votos positivos en la primera posición para así tener una respuesta referente al entrar a la duda en detalle.

3.4. CASOS DE USOS

El objetivo principal de este apartado es poder definir todas las especificaciones funcionales de la aplicación con suficiente detalle. De esta forma tener una orientación fija que nos pueda servir tanto para el desarrollo como para la verificación de que la aplicación se comporta de la forma esperada.

Hemos seguido el modelo de documentación de casos de uso adquirido en la asignatura del grado Diseño de Software. De esta forma, podemos extraer los requerimientos del actor (usuario) y del comportamiento del sistema, centrándonos en las tareas principales que hay en la interacción entre estos dos.

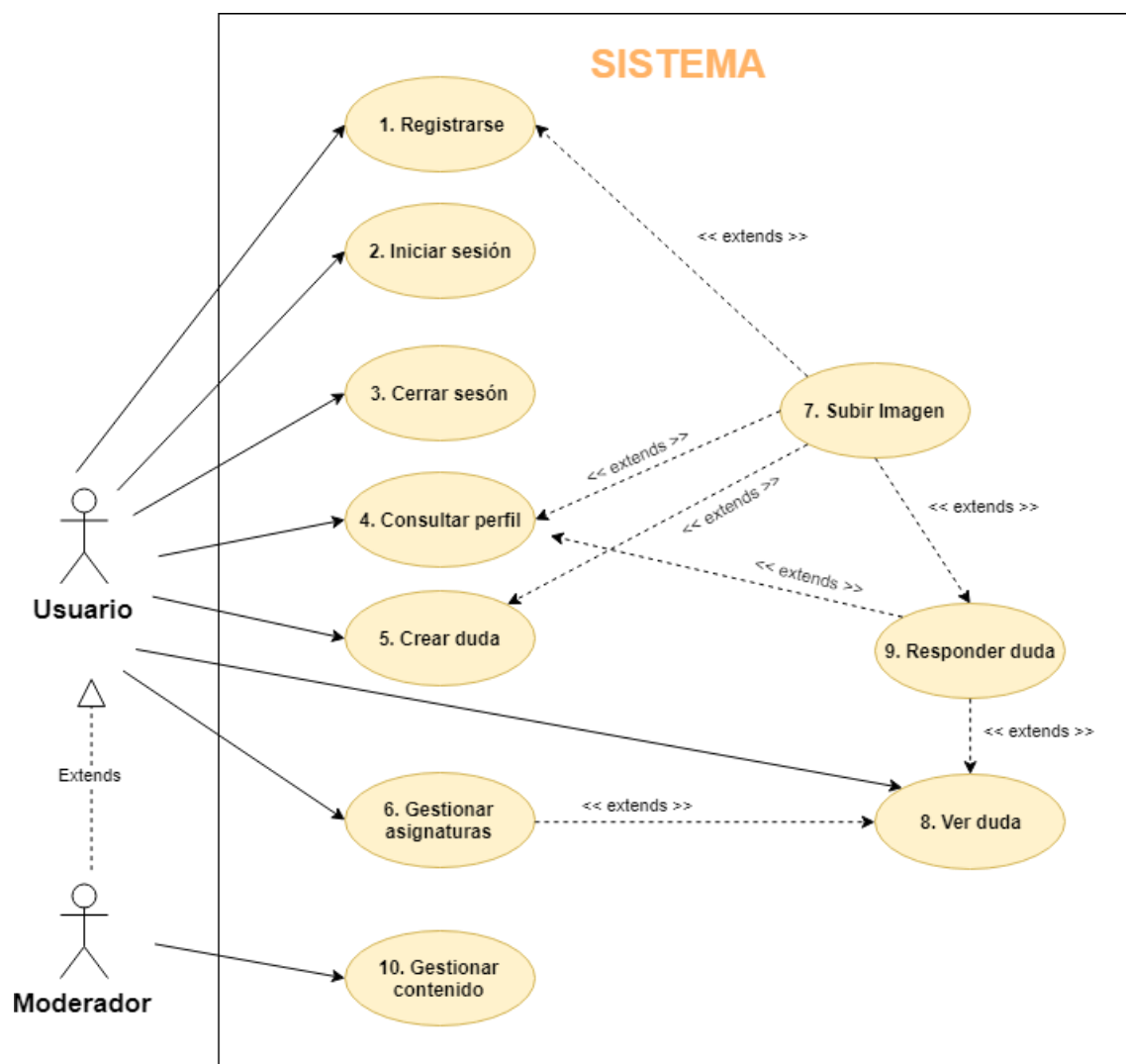


Figura 8: Casos de uso

A continuación, se especificarán cada uno de los casos de usos expuestos en el diagrama.

3.4.1. REGISTRARSE

Actor:	Usuario
Descripción:	Proceso por el cual el usuario se registra en la base de datos
Precondiciones:	Encontrarse en la pantalla inicial de la aplicación
Flujo básico: <ol style="list-style-type: none">1. El usuario selecciona la opción de “Registrarse”.2. El sistema muestra la pantalla de registro con el formulario a rellenar.3. El usuario introduce el nombre de usuario, correo electrónico y contraseña.4. El usuario selecciona la opción de registrar.5. El sistema comprueba los datos y los registra estos en la base de datos.6. El sistema muestra la pantalla principal la aplicación.	
Flujo alternativo: <ol style="list-style-type: none">3.a. El usuario puede seleccionar una imagen para su perfil.5.a. El sistema detecta datos incorrectos, muestra error en los campos en los que lo hay y no inicia sesión.	
Postcondiciones:	El usuario se encuentra en la pantalla principal de la aplicación

3.4.2. INICIAR SESIÓN

Actor:	Usuario
Descripción:	Proceso de autenticación de los datos del usuario en la base de datos
Precondiciones:	Encontrarse en la pantalla inicial de la aplicación
Flujo básico: <ol style="list-style-type: none">1. El usuario introduce su dirección de correo electrónico o el nombre de usuario y su contraseña.2. El usuario selecciona la opción de “Iniciar sesión”.3. El sistema comprueba los datos en la base de datos e inicia la sesión del usuario.4. El sistema muestra la pantalla principal la aplicación.	
Flujo alternativo: <ol style="list-style-type: none">3.a. El sistema detecta datos incorrectos, muestra error en los campos en los que lo hay y no inicia sesión.	
Postcondiciones:	El usuario se encuentra en la pantalla principal de la aplicación

3.4.3. CERRAR SESIÓN

Actor:	Usuario
Descripción:	Proceso por el cual el usuario sale del sistema
Precondiciones:	Encontrarse en la pantalla de perfil de la aplicación
Flujo básico: <ol style="list-style-type: none">1. El usuario selecciona la opción “Cerrar sesión”.	

2. El sistema muestra mensaje de confirmación. 3. El sistema cierra la sesión y muestra la pantalla inicial de la aplicación.	
Flujo alternativo:	
2.a. El usuario rechaza la confirmación, el sistema se cierra este mensaje.	
Postcondiciones:	El usuario se encuentra en la pantalla de perfil de la aplicación

3.4.4. CONSULTAR PERFIL

Actor:	Usuario
Descripción:	Proceso por el cual el usuario navega por su perfil.
Precondiciones:	El usuario se encuentra en cualquier pantalla que contenga la barra de navegación diferente a la pantalla de perfil.
Flujo básico:	
1. El usuario selecciona la opción de perfil en la barra. 2. El sistema le muestra la foto de perfil, ver mis dudas, ayuda, cerrar sesión.	
Flujo alternativo:	
2.a. El usuario selecciona sobre la foto de perfil para modificar esta. 2.b. El usuario selecciona ver mis dudas y el sistema muestra todas las dudas realizadas por él. 2.c. El usuario selecciona ayuda y el sistema muestra detalles útiles de la aplicación. 2.d. El usuario selecciona cerrar sesión.	
Postcondiciones:	El usuario se encuentra en la pantalla de perfil

3.4.5. CREAR NUEVA DUDA

Actor:	Usuario
Descripción:	Proceso por el cual el usuario crea una nueva duda
Precondiciones:	El usuario se encuentra en la pantalla de crear nueva duda
Flujo básico:	
1. El usuario selecciona la opción crear nueva duda en la barra de navegación. 2. El usuario introduce el título, la descripción y la asignatura a la que pertenece la duda. 3. El usuario selecciona el botón de enviar. 4. El sistema crea la nueva duda y notifica al usuario. 5. El sistema borra todos los campos introducidos.	
Flujo alternativo:	
2.a. El usuario introduce algún archivo de tipo imagen. 3.a. El sistema bloquea la pantalla y muestra barra de progreso hasta que se añade la duda en base de datos.	
Postcondiciones:	El usuario se encuentra en la pantalla de la crear una nueva duda

3.4.6. GESTIONAR ASIGNATURAS

Actor:	Usuario
Descripción:	Proceso por el cual el gestiona sus asignaturas para ver las diferentes dudas que tiene una en concreto, añadir una nueva asignatura o eliminar alguna
Precondiciones:	El usuario se encuentra en cualquier pantalla que contenga la barra de navegación
Flujo básico: <ol style="list-style-type: none">1. El usuario selecciona la opción ver Mis asignaturas en la barra.2. El sistema muestra todas las asignaturas que el usuario tiene añadidas.3. El usuario selecciona una asignatura.4. El sistema realiza la acción escogida.	
Flujo alternativo: <ol style="list-style-type: none">3.a. El usuario mantiene presionado sobre una asignatura.<ol style="list-style-type: none">3.a.1. El sistema le da la opción de borrarla de su lista de asignaturas.3.b. El usuario abre una asignatura para consultar sus dudas.<ol style="list-style-type: none">3.b.1. El sistema muestra el listado de las dudas que tiene la asignatura seleccionada siempre que tenga.3.c. El usuario selección añadir duda.<ol style="list-style-type: none">3.c.1. El sistema muestra la pantalla de cursos.	
Postcondiciones:	El usuario se encuentra en la pantalla de Mis asignaturas

3.4.7. SUBIR IMAGEN

Actor:	Usuario
Descripción:	Proceso por el cual el usuario sube una imagen.
Precondiciones:	El usuario se encuentra en la pantalla de registro, crear foro, en mi perfil o en un foro
Flujo básico: <ol style="list-style-type: none">1. El usuario selecciona una de las tres opciones comentadas.2. El sistema muestra la pantalla seleccionada.3. El usuario selecciona subir foto.4. El sistema muestra el archivo subido con una miniatura.	
Flujo alternativo: <ol style="list-style-type: none">1.a. El usuario selecciona subir imagen en caso de estar en las tres pantallas.	
Postcondiciones:	El usuario se encuentra en la pantalla de registro, crear foro, en su perfil o en el foro.

3.4.8. VER DUDA

Actor:	Usuario
Descripción:	Proceso por el cual el usuario llega a ver una duda de forma detallada y sus comentarios
Precondiciones:	El usuario se encuentra en la pantalla principal o en la de una asignatura en concreto
Flujo básico: <ol style="list-style-type: none">1. El usuario selecciona un foro.2. El sistema muestra la pantalla del foro.	
Flujo alternativo:	
Postcondiciones:	El usuario se encuentra en la pantalla de una duda

3.4.9. RESPONDER DUDA

Actor:	Usuario
Descripción:	El usuario pone un comentario sobre la duda que haya en la duda
Precondiciones:	El usuario se encuentra en la pantalla de duda
Flujo básico: <ol style="list-style-type: none">1. El usuario selecciona “Responder”.2. El sistema abre una caja de texto.3. El usuario entra el texto.4. El sistema publica la respuesta debajo de la última que haya.	
Flujo alternativo: <ol style="list-style-type: none">2.a. Al lado de la caja de texto hay un botón para seleccionar respuesta avanzada. Dicha respuesta tiene la posibilidad de adjuntar una o más imágenes.	
Postcondiciones:	El usuario se encuentra en la pantalla de duda con su respuesta publicada

3.4.10. GESTIONAR CONTENIDO

Actor:	Moderador
Descripción:	Proceso por el cual el moderador gestiona el contenido de la aplicación
Precondiciones:	Estar dentro de la aplicación
Flujo básico: <ol style="list-style-type: none">1. El moderador gestiona contenido de foros.	
Flujo alternativo: <ol style="list-style-type: none">1.a. El moderador advierte o elimina al usuario si tiene un mal comportamiento.1.b. El sistema borra de base de datos al usuario.	
Postcondiciones:	La aplicación estará gestionada

3.5. MODELO DE DOMINIO

En este apartado concluimos el análisis de nuestro proyecto presentando el diagrama de dominio (Figura 9) del sistema. Este diagrama es utilizado para comprender todo lo que abarca la idea a desarrollar en la propia realidad física, sin tener que entrar en el ámbito de *software*. Es el punto de partida para el diseño de toda la aplicación.

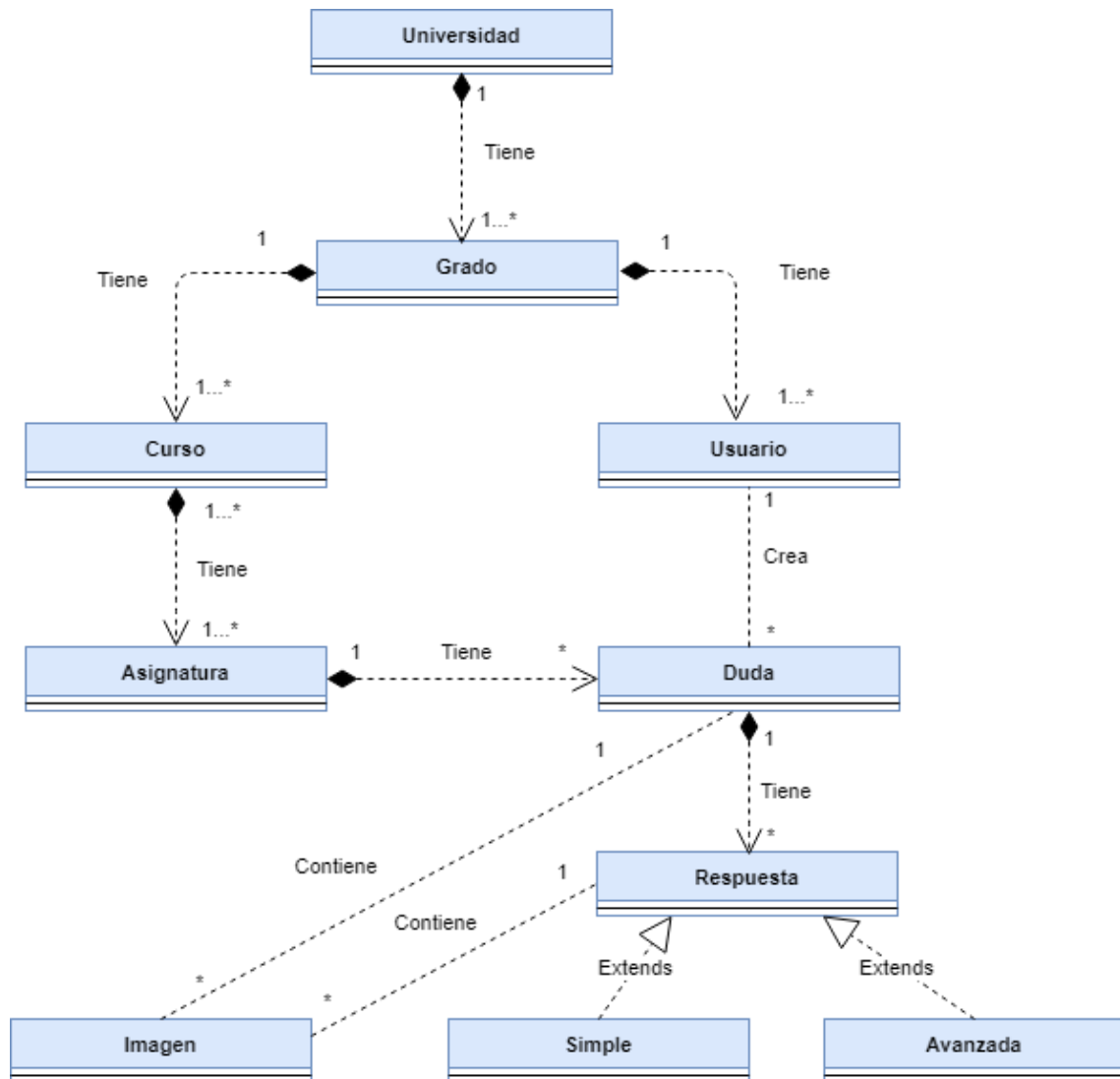


Figura 9: Modelo de dominio

Como podemos observar, tenemos un modelo que sigue una jerarquía donde la clase principal es la universidad y la supresión de esta conllevaría a la pérdida de todos los componentes que la constituyen. Esto es debido a que, si no existe la universidad, no existe el grado y, por lo tanto, ningún usuario/estudiante que pueda asistir a esta. De igual forma pasa con todas las clases que no son una composición fuerte, como lo son Imagen o Respuesta.

Hay que añadir que una respuesta puede ser simple o avanzada, donde sólo haya texto o existe la opción de subir una o más imágenes acompañando al texto. La creación de una nueva duda es equivalente en este aspecto, se pueden adjuntar imágenes.

4. PATRONES

A la hora de crear un *software*, hace falta tener una idea de ciertas características del proyecto a realizar. Por ende, hace falta examinar diferentes programas similares para así ayudarte a generar tu propia perspectiva ante dicho proyecto. Esto quiere decir que, ya otra persona ha podido estar ante una situación parecida y ha conseguido dar una solución a su obstáculo. Por lo que, si ya existe una solución a un problema, significa que existe un patrón de cómo resolver la duda. Esto es lo que se conoce como patrones de diseño, son herramientas útiles para tener en cuenta a la hora de programar para que así se facilite la reutilización de arquitecturas de *software* aprobadas por diferentes compañeros del campo laboral. Nosotros nos hemos basado en utilizar diferentes patrones, tanto para el diseño de interfaz gráfica y el diseño de programación.

4.1. DISEÑO DE INTERFAZ GRÁFICA

Para diseñar la interfaz gráfica de la aplicación hemos seguido y aplicado patrones de diseño web (Material Design 2018). El uso de estos patrones ayuda a mejorar la usabilidad y navegabilidad de la aplicación. Los diferentes patrones utilizados en este trabajo han sido:

- **Estados vacíos:** Estos estados se producen cuando no se puede mostrar el contenido. Por lo que evitar la pantalla en blanco y mostrar un mínimo de información mientras obtiene el contenido.
- **Iconos:** Utilizar iconos⁸ que hagan referencia a los objetos de una forma definida por la sociedad. De esta forma el usuario tiene facilidad de comprensión por su familiaridad con ellos.
- **Navegación:** La navegación guía a los usuarios a través de diferentes partes de la aplicación. Se debe organizar la estructura de la aplicación de acuerdo con las tareas principales para que el usuario las pueda tener a mano y, además, las pantallas deben de ser intuitivas y predecibles. Los usuarios nuevos deben ser capaces de descubrir cómo moverse a través de la aplicación con facilidad.

Dentro de este patrón de diseño hay diferentes sub-patrones. A continuación, destacamos los utilizados:

- **Iluminar acciones:** Destacar los elementos específicos iluminando más la parte *foreground*, la cual se va a utilizar y dejando el *background* de secundario.
- **Botón hacia atrás:** Este botón devuelve al usuario a la pantalla anterior que estaba. Navega hacia arriba en la jerarquía de la aplicación hasta que se alcanza la pantalla de inicio.
- **Barra de navegación inferior:** Permite a los usuarios moverse rápidamente entre un pequeño número de vistas de nivel superior. En nuestro caso las pantallas principales.

⁸ Herramienta de iconos: <https://romannurik.github.io/AndroidAssetStudio/>

- **Permisos:** Las solicitudes de permisos deben ser simples, transparentes y comprensibles. La aplicación debe aclarar por qué se necesita cada solicitud de permiso. En este caso permisos de cámara y almacenamiento.
- **Selección:** La selección permite indicar al usuario que ha seleccionado un elemento mediante una capa de color sobre él.
- **Confirmación y reconocimiento:** Cuando un usuario invoca una acción en la aplicación, que pueda reconocer esa acción a través de texto y confirmarla.

4.2. DISEÑO DE PROGRAMACIÓN

Para el diseño del proyecto, orientado al desarrollo de la lógica, nos hemos basado en implementar todo el código utilizando el patrón de diseño Modelo-Vista-Controlador (MVC). Este patrón de arquitectura de *software* separa los datos y la lógica de negocio de la interfaz de usuario (Figura 10). MVC es algo que aprendimos en el grado, por lo que ya estamos familiarizados a trabajar con él y resulta más cómodo poder dividir el proyecto de esta forma. No obstante, por temas del funcionamiento asíncrono y por la forma en la que trabaja la plataforma Firebase, hemos tenido que adaptarnos y realizar modificaciones del patrón respecto a cómo lo hemos utilizado a lo largo del grado.

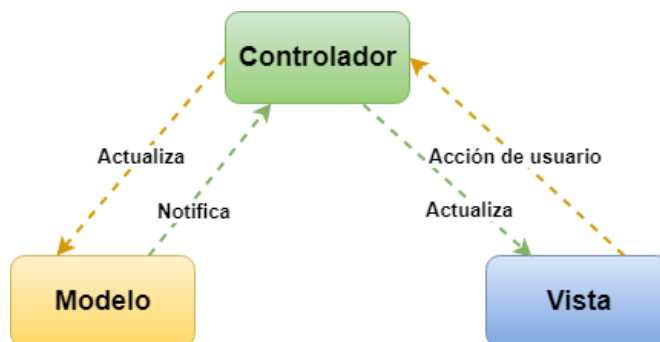


Figura 10. Patrón Modelo-Vista-Controlador

A continuación, exponemos qué engloba cada uno de estos componentes en nuestra aplicación:

- **Modelo:** Engloba todas las clases relacionadas con los objetos de nuestro modelo de dominio. Cada una de estas clases tiene su vinculación con la base de datos.
- **Controlador:** Engloba sólo la clase del controlador y dicha clase está basada en el patrón de diseño Singleton. De esta forma se asegura tener sólo una instancia de la clase para las diferentes clases que la utilizan. Esta clase se encarga de conectar con la base de datos a partir de los datos que ha obtenido desde vista y con ello crear los objetos de modelo que se envían a ésta. Además, proporciona métodos compartidos entre clases, siempre que las APIs de Firebase o Android lo permitan. Ya que múltiples

métodos (seleccionar imagen o abrir cámara) deben de ser invocados directamente desde su propia *Activity*.

- **Vista:** Engloba todas las clases relacionadas con las pantallas que el usuario interacciona en la aplicación, son las denominadas *Activities*. También se encarga de comprobar que los campos entrados por el usuario sean los correctos. Tiene la particularidad que puede conectarse con base de datos para obtener información desde Firebase para así realizar acciones que el usuario ejecuta. Entre estas acciones está iniciar sesión.

Para realizar la implementación de navegabilidad entre las principales pantallas, hemos observado que el patrón de Navegación de Fragmentos es el idóneo para la aplicación. Para comprender este patrón, hace falta saber que un fragmento es una sección modular de la interfaz de usuario embebida dentro de una actividad, por lo cual esto permite versatilidad de diseño y ahorrarse repetir parte de diseño como puede ser en nuestro caso, la barra de navegación. Este patrón ofrece un mejor rendimiento ya que no se debe ir creando una actividad cada vez que se abren las pantallas principales y, por consiguiente, hay un menor gasto de memoria.

Por último, para mostrar los diferentes datos como son las dudas, las respuestas o las imágenes en los respectivos contenedores RecyclerView (Developers Android 2018) que tiene la aplicación, hemos utilizado el patrón ViewHolder (Quinzá Pérez, C. 2016) para así tener una usabilidad ágil y fluida. Este patrón consiste en almacenar la referencia a los elementos que tienen en común los datos del diseño que se van a ver por pantalla y, por lo tanto, ahorra ir a buscarlos cada vez que van a ser visualizados.

5. DESARROLLO Y RESULTADOS

Como hemos comentado, hemos realizado el desarrollo de la aplicación separándolo en cuatro incrementos, donde cada uno abarca el incremento anterior implementado. Con esta separación conseguimos poder comenzar a construir toda la idea comenzando desde lo más asequible hasta llegar a nuestro objetivo final: una aplicación estable que permita compartir y resolver dudas entre los compañeros.

En este apartado expondremos el desarrollo para cada incremento, con la especificación de éste como introducción, los requisitos funcionales y especificaciones para tener en cuenta de lo que se va a implementar. Además, explicaremos el diagrama de clases que se ha realizado y también se detallarán aspectos destacados para cada una de las clases que hay en éste. Comentar que se añadirá, siempre que sea necesario, partes de código que creemos importantes mostrar por su valor dentro del desarrollo o para aclarar lo explicado. En cada incremento no incorporaremos clases de diagramas anteriores, a no ser que haya habido una modificación en alguna de las clases en el incremento actual. A partir del segundo incremento, hemos realizado una evaluación de la aplicación para así modificar la aplicación dependiendo de la opinión de diferentes usuarios. También detallaremos el comportamiento de la aplicación, a la par que mostramos el diseño final de esta. Este subapartado es muy útil para ver los resultados de cada incremento y comprender cómo funciona la aplicación. Por último, expondremos las pruebas realizadas para así comprobar que hemos cumplido con los requerimientos deseados. Estas pruebas se han realizado de forma manual con diferentes emuladores tratando de entregar un *software* robusto.

5.1. INCREMENTO 1: INICIO DE SESIÓN Y REGISTRO

Es el Incremento básico de la aplicación y el más importante, ya que es lo que el usuario verá al abrir la aplicación y, por lo tanto, debe impresionar para que así quiera navegar por ésta. Es fundamental que el usuario pueda formar parte de la comunidad de una forma que le resulte rápida, simple e intuitiva.

5.1.1. REQUISITOS FUNCIONALES Y ESPECIFICACIONES

1. El usuario debe tener una cuenta para utilizar la aplicación. Esta cuenta se realiza en la pantalla de registro y se debe rellenar la siguiente información:
 - Nombre de usuario: El nombre que utilizará en la comunidad tiene que ser único, por lo que se comprobará en la base de datos su existencia y también deberá tener mínimo 4 caracteres para que sea válido.
 - Correo electrónico: Debe tener el formato alfanumérico, seguido por el signo @ y finalizado por un nombre de dominio. Se comprobará también de su existencia en la base de datos.

- Contraseña: Debe contener al menos 6 caracteres y puede observarla para así evitar que la repita. Para esto último, hace falta que exista un botón que habilita y deshabilite la opción de ver contraseña.

- Foto (opcional): Puede subir una imagen o capturar una al presionar sobre el icono de la cámara que hay en la pantalla de registro. Una vez presionado este icono, saldrá una ventana emergente para confirmar si sube de galería o realiza una fotografía desde su cámara. Una vez escogida una imagen, sin importar la opción escogida anteriormente, esta imagen se pondrá visible donde anteriormente estaba el icono de la cámara. Si no se selecciona ninguna imagen, tendrá por defecto una. La cual podrá modificar en un futuro cuando haya pantalla de área personal.

2. El usuario puede iniciar sesión mediante el correo electrónico o nombre de usuario registrado y su contraseña.

3. Una vez el usuario haya creado la cuenta correctamente o haya iniciado sesión, se encontrará en la pantalla principal de la aplicación (“Últimas dudas”).

4. El usuario puede recuperar su contraseña ingresando el correo electrónico dentro de la caja de texto de la ventana emergente. Recibirá un correo con las indicaciones para reestablecer la contraseña.

5. El usuario debe de estar siempre informado de sus fallos, por lo que se generarán mensajes de error determinados.

5.1.2. DIAGRAMA DE CLASE

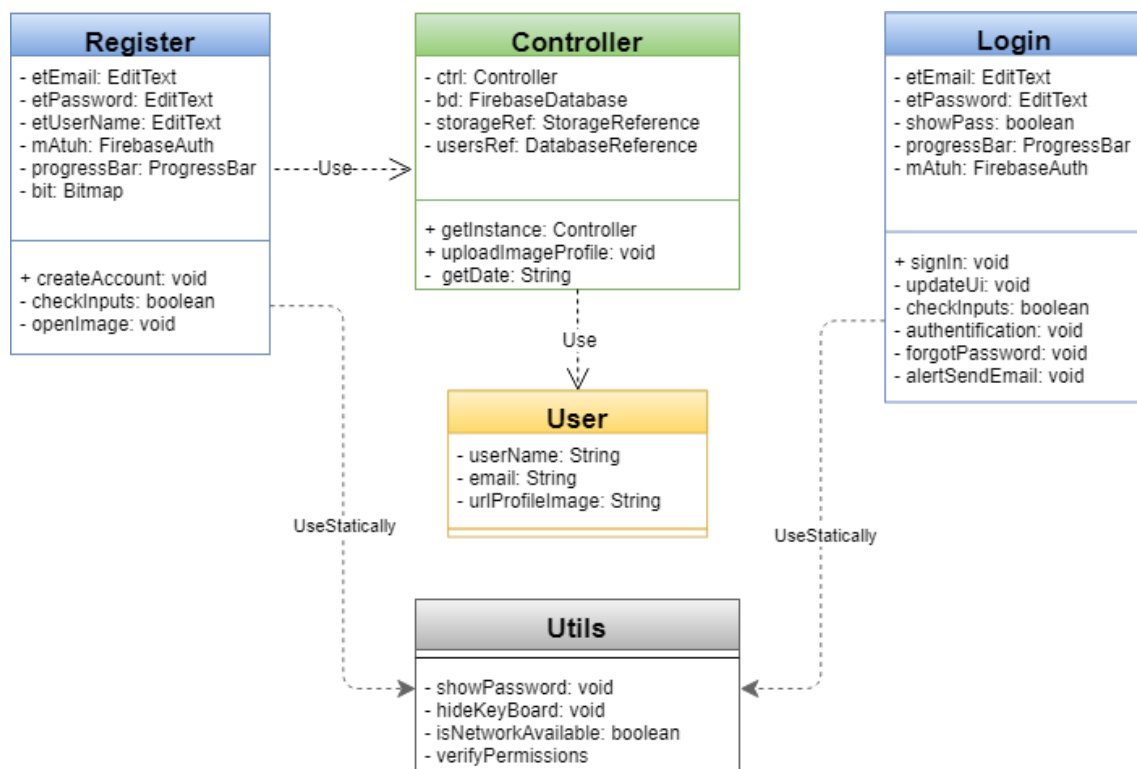


Figura 11. Diagrama de clases del incremento 1 correspondiente al modelo MVC

Como se puede observar, hemos separado las clases respecto al diseño que le corresponde en el modelo MVC. Donde el color naranja hace referencia a Modelo, el azul a Vista y el verde a Controlador. También hemos puesto los detalles más importantes de cada una de estas clases. A continuación, expondremos los aspectos del diagrama más importantes:

- **User:** Clase de modelo que contiene toda la información básica del usuario. Al añadir un objeto de este tipo a la base de datos se genera la tabla de la Figura 12 donde Firebase crea un uid único para cada nuevo elemento que se añada. Cabe aclarar que no es necesariamente obligatorio poner el uid que te genera la plataforma como el identificador del usuario, pero dicho uid facilita la obtención de un usuario a la hora de administrarlos ya que la librería FirebaseAuth te da la opción de obtenerlo partir de un uid.



Figura 12. Tabla de usuarios

- **Login:** Es la clase inicial de la aplicación y pertenece al conjunto de vista, pero funcionando de una forma híbrida. Muestra diferentes posibilidades al usuario como iniciar sesión, recuperar contraseña o registrarse. Los métodos más importantes son los siguientes:
 - **SignIn:** Es el método que se encarga de obtener desde base de datos el usuario que quiere iniciar sesión, siempre y cuando los datos introducidos coincidan con alguno ya registrado. Si todo va bien, llama al método `updateUI` para abrir la aplicación (pantalla de inicio), sino muestra error.
 - **UpdateUI:** Es un método destacado ya que se encarga de abrir la aplicación si es la primera vez que el usuario inicia sesión, pero también, es capaz de comprobar para cada dispositivo si un usuario ya había iniciado sesión anteriormente en dicho dispositivo. Esto quiero decir que evitará pasar por la pantalla de “Login” al usuario y de esta forma ahorrarle tener que entrar los datos para iniciar sesión cada vez que entra en la aplicación.
- **Register:** Clase también de vista que proporciona al usuario un pequeño formulario a rellenar para así poder formar parte de la comunidad. De esta clase destacan los siguientes métodos:
 - **CreateAccount:** Si los datos recogidos en el formulario son correctos y la cuenta de usuario a registrar no se encuentra ya en la base de datos, crea la cuenta de usuario y después envía este nuevo usuario al controlador para que ahí se añada la foto de perfil en el *storage* de Firebase.
 - **OpenAlert:** Este método se encarga de mostrar al usuario una ventana emergente (*pop-up*) con las opciones de subir una imagen desde galería o realizar una fotografía desde cámara después de haber aceptado los permisos solicitados.

- **Controller:** Única clase de controlador que realiza la creación de los objetos de modelo en la base de datos. Al ser un *singleton*, el método base de esta aplicación es el `getInstance`, donde se obtienen, sólo una vez, todas las referencias de la base de datos que necesitamos usar para diferentes acciones y el usuario que ha iniciado sesión. Con esto último conseguimos tener una interacción con el actual usuario mucho más cómoda al tener a mano todos sus atributos. En la Figura 13, podemos observar el método de inicializar los elementos al crear una instancia de esta clase y también el evento que se realiza a una referencia específica (usuarios) para así obtener un dato desde base de datos, en este caso el usuario.

```
private void initElements() {
    db = FirebaseDatabase.getInstance();
    storageRef = FirebaseStorage.getInstance().getReference();
    usersRef = db.getReference(Constants.REF_USERS);
    doubtsRef = db.getReference(Constants.REF_DOUBTS);
    subjectsRef = db.getReference(Constants.REF_SUBJECTS);
    coursesRef = db.getReference(Constants.REF_COURSES);

    usersRef.child(FirebaseAuth.getInstance().getCurrentUser().getUid()).addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {
            user = dataSnapshot.getValue(User.class);
        }

        @Override
        public void onCancelled(DatabaseError databaseError) {
        }
    });
}
```

Figura 13. Obtención de las referencias y del usuario actual de la aplicación

- **Utils:** Esta clase no corresponde a ningún conjunto de MVC, pero es utilizada por las clases de vista, tanto Register como Login. Lo que pretendemos es tener en una clase métodos utilizados por ambas. Entre estos métodos tenemos la posibilidad de: mostrar la contraseña, saber si no disponemos de conexión a la hora de registrarnos o iniciar sesión, esconder el teclado si se presiona fuera de éste y verificar los permisos del dispositivo en la pantalla de registro para poder añadir una imagen mediante la galería o una hecha al instante con la cámara.

5.1.3. COMPORTAMIENTO

Al abrir la aplicación se muestra la pantalla para iniciar sesión (Figura 14A). El inicio de sesión puede realizarse mediante el nombre de usuario o correo electrónico y la contraseña que se registró. Se puede observar que al lado de la contraseña hay un ojo, el cual se puede presionar para así hacerla visible. Este ojo cambia de color a amarillo si la contraseña está visible, si no, vuelve al estado por defecto.

Una vez entrados los datos, se debe presionar sobre el botón 'INICIAR SESIÓN' y si los campos son incorrectos o simplemente están vacíos, se mostrará un mensaje de error notificando en cada campo (Figura 14B). En caso de que estos campos estén correctos, se mostrará al usuario la pantalla de "Últimas dudas". Además de permitir iniciar sesión, esta pantalla tiene la opción de cambiar la contraseña en caso de haberla olvidado. Para llevar a cabo esta función, se debe presionar sobre el texto: '¿Has olvidado contraseña?'. Seguidamente se abre una pequeña ventana (Figura 14C) que permite entrar el correo electrónico del cual se ha olvidado la contraseña y luego de dar a 'Enviar', se notificará al usuario (Figura 14D) si ha sido posible realizar dicha acción, ya que el correo entrado puede no existir en la base de datos. Si es un correo electrónico registrado, el usuario deberá seguir los pasos que se indican en el mensaje que le llegará al correo entrado y con esto podrá reestablecer su contraseña. Las notificaciones al usuario, es un tipo de mensajes generados para múltiples acciones dentro de la aplicación además de ésta. Es una forma sencilla donde de informar al usuario, donde obstruye una pequeña parte de la pantalla y desaparecen a los pocos segundos. Este tipo de mensajes se denominan *SnackBar*. Otra característica de la pantalla es que da la opción de ir a la pantalla de registro, también presionando sobre el texto, por si el usuario todavía no se dispone de una cuenta.

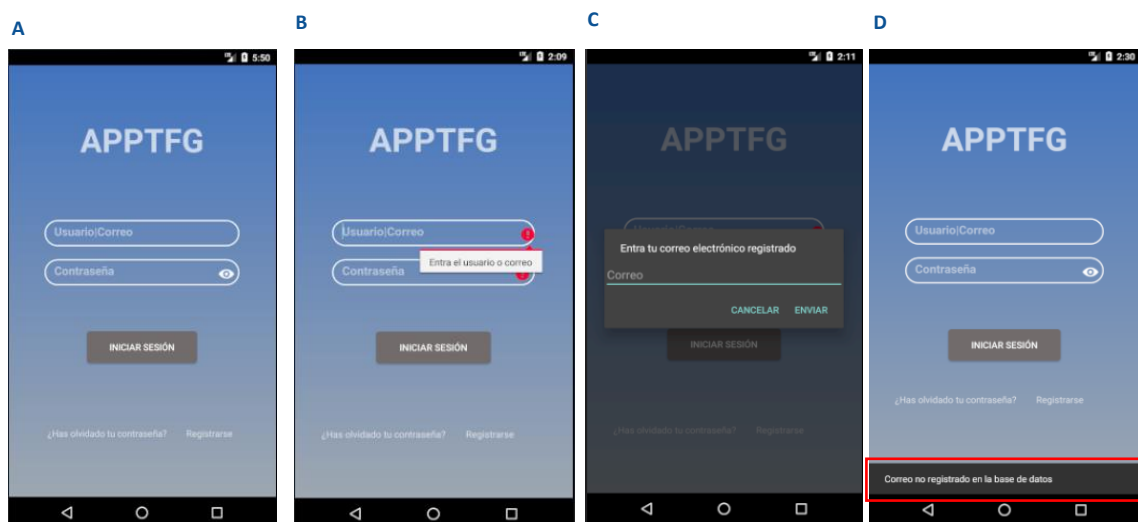


Figura 14 . Pantalla de Login

En la pantalla de registro (Figura 15A) se puede observar que hay diferentes campos (obligatorios) para rellenar y la posibilidad de añadir una imagen de perfil si se presiona sobre el icono del cámara situado en la parte superior. Una vez presionado, se pedirá aceptar los permisos necesarios para poder acceder a la galería y para acceder a la cámara. Después de aceptar estos permisos (Figura 15B), se deberá presionar sobre el icono nuevamente y se abrirá una ventana (Figura 15C) que da la opción a escoger entre seleccionar una imagen de la galería o abrir la cámara para realizar una imagen en ese momento. Si no se desea añadir ninguna foto de perfil, se asigna una imagen por defecto. De la misma forma que la pantalla anterior, todos los campos notificarán mediante un mensaje de error si ha habido algún problema, justo después de haber presionado sobre el botón 'CREAR CUENTA'. Si todo ha ido correctamente, se abrirá la pantalla de "Últimas dudas".

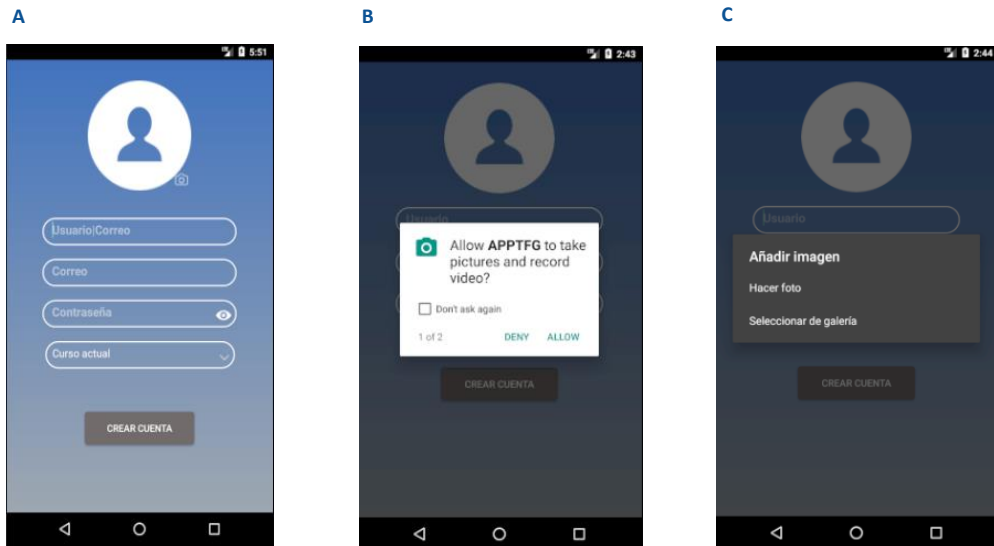


Figura 15. Pantallas de registro

5.1.4. PRUEBAS DEL FUNCIONAMIENTO

Para poder comprobar que la implementación de la fase ha sido la correcta, realizamos diferentes pruebas:

1. Comprobar que se den correctamente los diferentes errores para todos los campos de texto.
2. Registrarse con un nombre de usuario ya existente en la base de datos.
3. Registrarse con un correo electrónico ya existente en la base de datos
4. Intentar iniciar sesión con un usuario y una contraseña no registrados.
5. Reestablecer la contraseña.
6. Iniciar sesión mediante usuario y también por correo electrónico.
7. Comprobar que todos los datos estén correctamente en los apartados de autenticación, base de datos y *storage* dentro de la plataforma Firebase.

5.2. INCREMENTO 2: FORO BÁSICO

Incremento base que cubre el objetivo principal de la aplicación, la creación de una aplicación con diferentes foros donde se puedan debatir dudas. Es necesario realizar un correcto desarrollo de esta Incremento ya que una vez has iniciado sesión, la aplicación depende de dicha herramienta para poder compartir las dudas.

5.2.1. REQUISITOS FUNCIONALES Y ESPECIFICACIONES

1. El usuario debe haber iniciado sesión para poder observar y crear los foros. Una vez dentro, la primera pantalla que se mostrará será la de “Últimas dudas” - ver Figura 20A - donde se pueden observar las últimas dudas añadidas en orden cronológico.

2. Para que haya navegabilidad entre las diferentes pantallas de la aplicación hace falta tener una barra de navegación para así poder ir a la pantalla de crear un nuevo foro o ver todos los existentes.

3. La barra de navegabilidad estará situada en la parte inferior de la pantalla y tendrá iconos referentes a las diferentes pantallas principales de la aplicación. Son las siguientes:

- Últimas dudas: Esta pantalla debe tener las dudas más recientes con los detalles que contienen las “cajas” de dudas.
- Mis asignaturas: Esta pantalla se desarrollará en la Incremento 3.
- Crear duda: Pantalla que permite crear una nueva duda.
- Mi perfil: Esta pantalla se desarrollará en la Incremento 4.

4. La “caja” de dudas tendrá los siguientes detalles:

- Día en que se posteo.
- Botón para dar “me gusta” y “no me gusta”, junto al número total que tiene cada uno.
- Número total de respuestas que tiene cada foro.
- Foto de perfil en circular al lado del nombre de usuario, tanto en la duda como en la respuesta.
- Título de la duda y debajo la descripción de ella.

5. Para crear una nueva duda se debe rellenar la información que se muestra a continuación:

- Título de la duda: con un mínimo de 5 y un máximo 30 caracteres.
- Descripción: Mínimo de 8 caracteres detallando la duda que se tiene.
- Seleccionar la asignatura a la cual pertenece la duda mediante un *Autocomplete*. Este punto se desarrollará en la Incremento 3.
- Posibilidad de adjuntar una o varias imágenes que ayude a describir la duda.

6. Para añadir una respuesta a una duda se debe entrar algún carácter y debe haber el botón para enviarla.

7. Las respuestas deben de estar ordenadas por el número de “me gustas” (teniendo en cuenta la diferencia con los “no me gusta”) que tiene. Poniendo así en primera posición la respuesta más votada por los usuarios.

5.2.2. DIAGRAMA DE CLASE

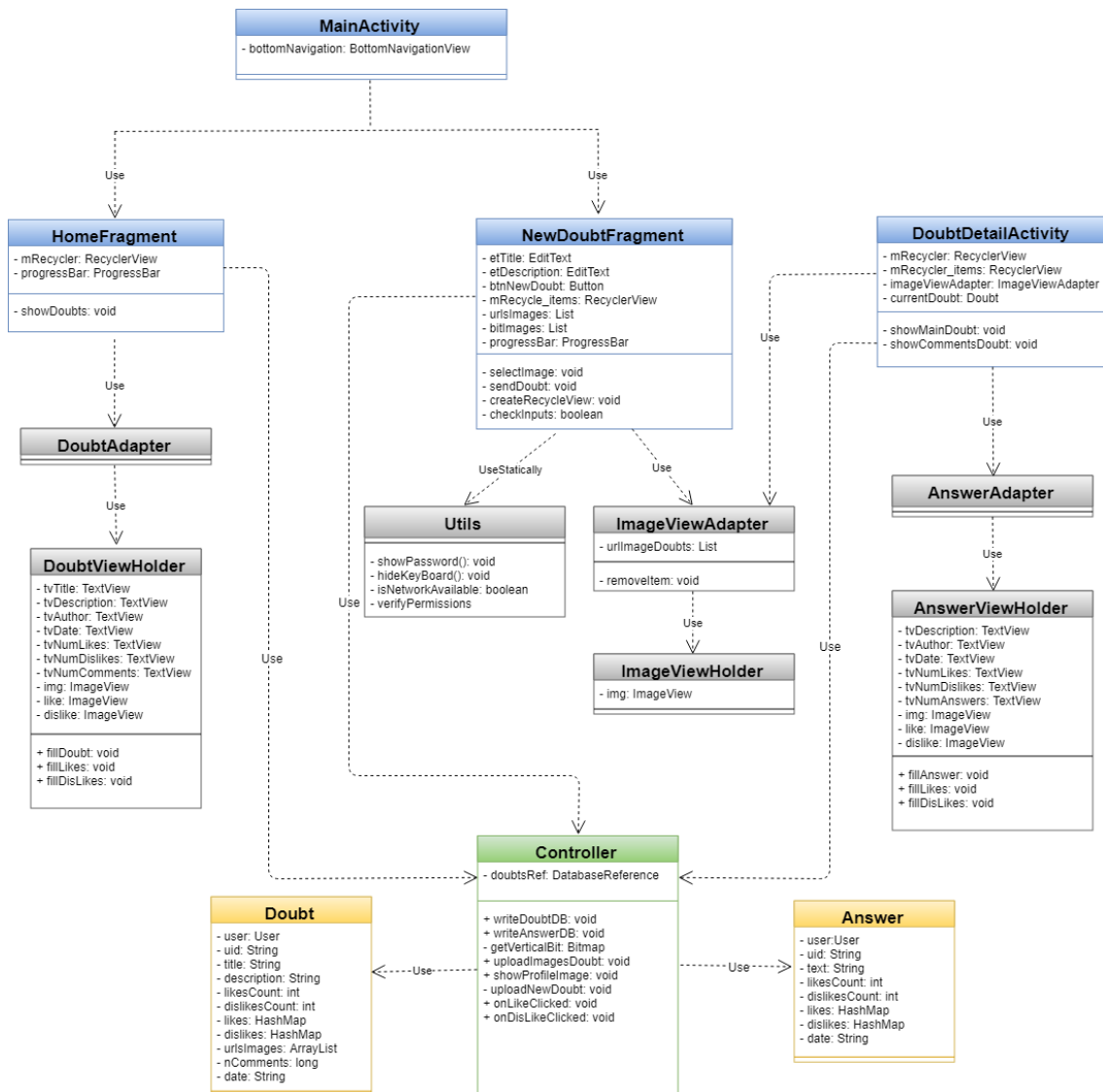


Figura 16. Diagrama de clases del incremento 2

En este diagrama también hemos separado las clases respecto al diseño que le corresponde en el modelo MVC, pero también hemos evitado repetir los datos del anterior diagrama de clases del incremento 1 para así centrarnos en la información que realmente se ha añadido en este nuevo incremento. Siempre conservando lo del anterior incremento. A continuación, expondremos los aspectos más importantes del diagrama:

- **MainActivity:** Esta clase es la *activity* principal que utilizan los *fragments*. Su función es mostrar la pantalla que corresponda según lo escogido en la barra de navegación.
- **HomeFragment:** Esta clase es el *fragment* principal que se carga por defecto al abrir la aplicación y dentro de él destaca lo siguiente:

- El objeto `mRecyclerView`: Es un contenedor de la clase `RecyclerView` que permite mostrar una lista de elementos, en este caso las dudas, de forma similar a una `ListView` pero con la ventaja que “recicla” los elementos que ya no son visibles por parte del usuario al hacer *scroll* en la pantalla. Esto optimiza la aplicación por no guardar todos los elementos y sólo mostrar los que realmente está observando el usuario.

Cabe destacar que para el uso del `RecyclerView`, por parte del diseño (xml), sólo es crear el elemento y añadir el tipo de ítems que se van a mostrar, Figura 17. De esta forma se indica que habrá una lista de elementos del XML desarrollado en `item_doubt`, que da como resultado una lista de cajas de la duda (Figura 18).

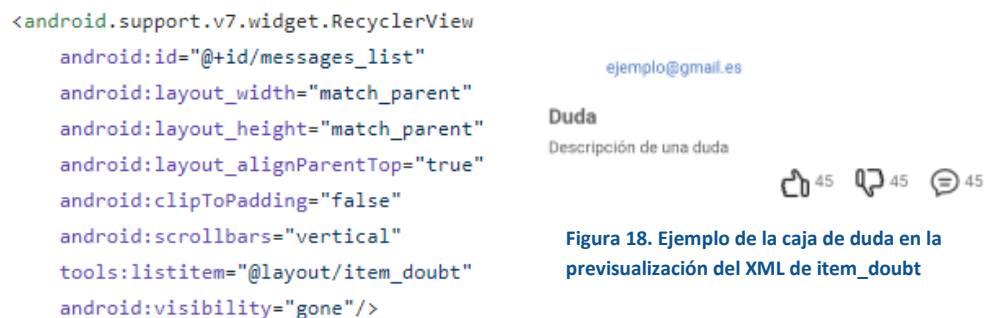


Figura 17. Código XML de `RecyclerView` para las dudas

- Método `showDoubts`: Se encarga de llamar a las funciones necesarias para así aplicar el patrón `ViewHolder` y mostrar todas las dudas por pantalla. Para beneficiarse aún más este patrón, hemos utilizado la clase `DoubtAdapter`, la cual extiende de `FirestoreRecyclerViewAdapter`, para así facilita toda la obtención y visualización de las dudas en tiempo real. Esta clase tipo *adapter* es proporcionada por `Firestore` y trabaja directamente con los objetos guardados en la base de datos, por lo que es capaz de gestionarlo de una forma fluida.
- **NewDoubtFragment:** Esta clase se encarga de comprobar todos los datos entrados por el usuario y de ser correctos, añadir una nueva duda a la base de datos. Aquí destaca el uso de la clase `ImageViewAdapter` que se utiliza para visualizar en un `RecyclerView` las imágenes seleccionadas a subir con la duda. Para mostrar estas imágenes hemos utilizado la librería `Glide` (Y.-J. Song, S.-b. Ou and J.-w. Lee (2016)) que nos proporciona mostrar la imagen en el elemento que deseamos a partir de una *url*. `Glide` también lo hemos utilizado para mostrar la foto de perfil que acompaña cada duda o respuesta de la aplicación. Además, para visualizar la imagen de la duda en grande, ya sea a la hora de crear una nueva o en la pantalla de “`DoubtDetailActivity`”, nos ha ayudado la librería `ImagePopup` (Image Popup (2017)) que realiza esta acción de una forma muy sencilla. Estas dos librerías hacen un gran aporte en pocas líneas como se puede ver en la Figura 19.

```
String url = urlImagesDoubt.get(position);
Glide.with(activity)
    .load(url)
    .thumbnail(0.1f)
    .into(viewHolder.img);
final ImagePopup imagePopup = new ImagePopup(activity);
imagePopup.initiatePopupWithGlide(url);
imagePopup.setFullScreen(true);
imagePopup.setImageOnClickClose(true);
```

Figura 19. Uso de las librerías Glide y ImagePopup

- **DoubtDetailActivity:** En esta clase se muestra la duda seleccionada en HomeFragment con toda su información y también las respuestas que se dan a ésta. Es la clase donde se gestiona todo el foro.
- **Controller:** Los nuevos métodos destacados de esta clase son:
 - **WriterDoubtDB:** Si los datos recogidos en el formulario que ofrece NewDoubtFragment son correctos, se crea una nueva duda. Esta creación cambia según si la duda tiene imágenes o no. En el caso de que disponga, primero se suben las imágenes de la duda y después se crea la duda con las *urls* de estas porque así se facilita y se agiliza el mostrar todas las imágenes haciendo uso de Glide. En el caso contrario, se crea directamente la duda.

5.2.3. EVALUACIÓN

Con una aplicación base, la cual permite la creación de dudas y poder responder a estas, se realizó una evaluación con usuarios reales para así obtener un *feedback* sobre la aplicación y que ellos pudieran participar en el diseño aportando nuevas ideas para así después aplicarlas en la aplicación.

La evaluación se ha hecho con un total de cuatro usuarios, de los cuales tres tienen una gran experiencia en aplicaciones móviles y el restante normal. El método de utilizado ha sido entregándoles la aplicación para que la testearan tanto como pudieran. Con la instalación de la aplicación en diferentes móviles, también se ha comprobado el funcionamiento de ésta en diferentes versiones Android y *hardware*. Las ideas que sugirieron los usuarios con los que se hicieron pruebas, son las siguientes:

- En la pantalla últimas dudas no mostrar toda la descripción de la duda, sólo una parte y ya al clicar sobre la duda que se abra la pantalla con la descripción total y sus respuestas.
- Previsualización de las imágenes de la duda a la hora de cargar éstas para así dar efecto de descargando mientras se obtienen del servidor.
- Añadir la hora y no sólo la fecha en la que se puso la duda o el comentario.
- Poder visualizar más líneas a la hora de escribir una respuesta sin necesidad de hacer *scroll*.
- Añadir fondo blanco en forma de cajas para los componentes de entrada en la pantalla de crear una nueva duda para diferenciar los campos.

5.2.4. COMPORTAMIENTO

Al abrir la aplicación se muestra directamente la pantalla de Últimas dudas, siempre y cuando se haya iniciado sesión anteriormente en ese dispositivo móvil. En esta pantalla (Figura 20A) se pueden ver todas las dudas que se han realizado por los usuarios. Consta de diferentes detalles: el nombre y la foto de perfil del usuario que ha subido su duda, el título y descripción de ésta, la fecha en la que se realizó, botones con la posibilidad de dar “me gusta” y “no me gusta”, junto a estos botones el número total de usuarios que tienen activo dicho botón en ese momento y finalmente, el número de comentarios que tiene la duda.

Una vez se presiona sobre una duda, se abre una nueva ventana (Figura 20B) con la duda y las mismas características comentadas anteriormente, pero con la descripción al completo, las imágenes (en miniatura) que se han adjuntado a ella (siempre y cuando tengan) y las respuestas que se le han dado diferentes usuarios a dicha duda.

Si una duda tiene diferentes imágenes, se pueden recorrer las miniaturas deslizando el dedo de forma horizontal para así verlas todas. Si el usuario presiona sobre cualquier imagen, se abrirá la imagen en tamaño completo para así observarla mejor y si se presiona otra vez sobre la pantalla, se cerrará.

Las respuestas que han dado a la duda también tienen un formato similar a la duda, pues consta de todas las características que ésta, pero eliminando el campo de número de comentarios total. Para poner una respuesta, hace falta escribir algún carácter en el *input* que hay en la parte inferior de la pantalla y presionar sobre el botón ya característico en muchas aplicaciones. Dicho botón se deshabilita una vez es presionado para así evitar enviar múltiples respuestas en caso de disponer una mala conexión en ese momento.

Para poder tener dudas en la pantalla comentada anteriormente, hace falta tener la posibilidad de crear estas, por lo que para ello se necesita la pantalla de “Nueva duda”. En esta pantalla (Figura 20C), se puede ver que para crear una duda se necesitaría rellenar los campos de título y descripción, estos campos tienen un mínimo de caracteres para así tener “obligar” al usuario detallar la duda con un mínimo de información. En el caso del título hay un máximo de caracteres para así tener títulos con palabras claves y concisas. También hay la posibilidad de añadir imágenes a la duda para así dar soporte a ésta.

Si se presiona sobre el icono de la cámara, se solicitan permisos (si todavía no se han aceptado) y luego se abre una pantalla para seleccionar una o diversas imágenes de multimedia. Existe un máximo de poder adjuntar un total de 8 imágenes por duda. Una vez seleccionadas las imágenes a subir con la duda, estas aparecen en la pantalla con la posibilidad borrar cualquiera de ellas si se mantiene presionándola durante 2 segundos. Una vez pasado este tiempo, se muestra una ventana (Figura 20D) para que el usuario confirme la acción. Siempre se puede añadir una nueva imagen después de haber añadido otras, ésta se pondrá en el contenedor de las imágenes (RecyclerView).

La opción de adjuntar imágenes es el único campo no obligatorio a la hora de presionar sobre el botón 'CREAR DUDA'. Este botón y los diferentes componentes de la pantalla se deshabilitan, de la misma forma que con el botón de enviar, con el fin de evitar múltiples creaciones de dudas y que el usuario no pueda modificar los elementos si no se dispone de una buena conexión. Una vez se ha subido la duda, sale un mensaje de confirmación, se borran todos los datos entrados y se habilitan todos los componentes para realizar una nueva duda.

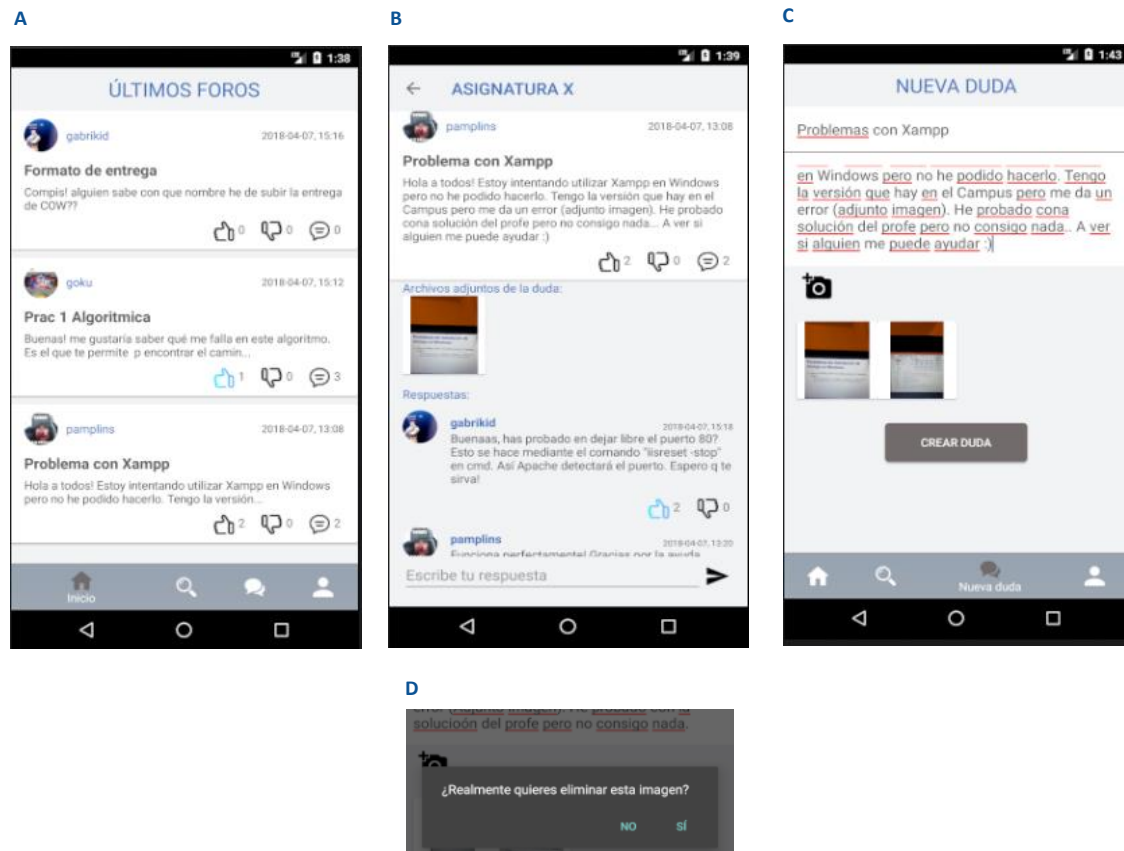


Figura 20. Pantallas destacadas del incremento 2

5.2.5. PRUEBAS DEL FUNCIONAMIENTO

- Entrar campos vacíos a la hora de crear una nueva duda o enviar una respuesta
- Crear una duda sin internet, tanto con imágenes como sin ellas.
- Ver dudas y comentarios sin conexión a internet. Estar navegando por la aplicación y perder el internet no impide seguir observando la aplicación con lo descargado de la base de datos y es posible realizar una nueva duda o nueva respuesta y los datos se actualizan una vez vuelve la conexión.
- Conectar múltiples dispositivos a la hora y ver que todo funciona correctamente.
- Ver que en la base de datos y también en diferentes dispositivos se actualizan los datos, como pueden ser: nuevo comentario, botones de “me gusta” y “no me gusta” y nueva duda.

5.3. INCREMENTO 3: CURSOS Y ASIGNATURAS

Con una aplicación que permite compartir dudas y debatir sobre estas, este incremento se centra en distribuir y organizar cada uno de estos foros en la asignatura que le corresponde. Esto da la posibilidad, a cada uno de los usuarios de la comunidad, a poder gestionar sólo las asignaturas que en ese mismo instante estén cursando o tenga motivación por ayudar en un tema que le resulta cómodo.

5.3.1. REQUISITOS FUNCIONALES Y ESPECIFICACIONES

1. El usuario debe haber iniciado sesión para poder ver y añadir asignaturas. Una vez dentro, la primera pantalla que mostrará será la de “Últimas dudas”, donde seguirán mostrándose las “cajas” mencionadas en el incremento anterior más un detalle añadido a estas cajas: el nombre de la asignatura a la que pertenece la duda.

2. El usuario debe ser capaz de elegir a qué asignatura pertenece la duda en el momento de crear una nueva mediante un *Autocomplete*.

3. Al seleccionar la pantalla de “Mis asignaturas”, el usuario podrá añadir una nueva asignatura siguiendo el siguiente procedimiento:

- El usuario presiona el botón de añadir una nueva duda y se abrirá la pantalla de selección de curso, donde habrá cuatro cajas con el nombre del curso, una caja para las optativas y una caja con el nombre de ‘Otros’ donde se encuentra el TFG y Empresa.
- Una vez seleccionado un curso o caja, se mostrarán una nueva pantalla de selección de asignaturas donde habrá nuevas cajas con el nombre de las asignaturas relacionadas a dicho curso o caja seleccionada anteriormente. Lo que nos lleva al siguiente punto.
- Esta pantalla debe separar las asignaturas en el semestre que le corresponde y al presionar sobre una, cambiar a color verde para así notificar al usuario que la tiene seleccionada. Se podrá presionar sobre una asignatura previamente seleccionada para así cancelar. Una vez finalizado, al presionar sobre el botón que se facilita, se llevará al usuario a la pantalla de “Mis asignaturas” con las nuevas asignaturas.
- Las asignaturas que el usuario ya tiene añadidas en la pantalla de “Mis asignaturas”, serán deshabilitadas en la pantalla de seleccionar asignaturas, con color naranja, para así no permitir que añada una que ya tiene.

4. La caja de la asignatura debe contener:

- Nombre de la asignatura.
- Curso y semestre al que pertenece.
- Número de dudas en total.

5.3.2. DIAGRAMA DE CLASE

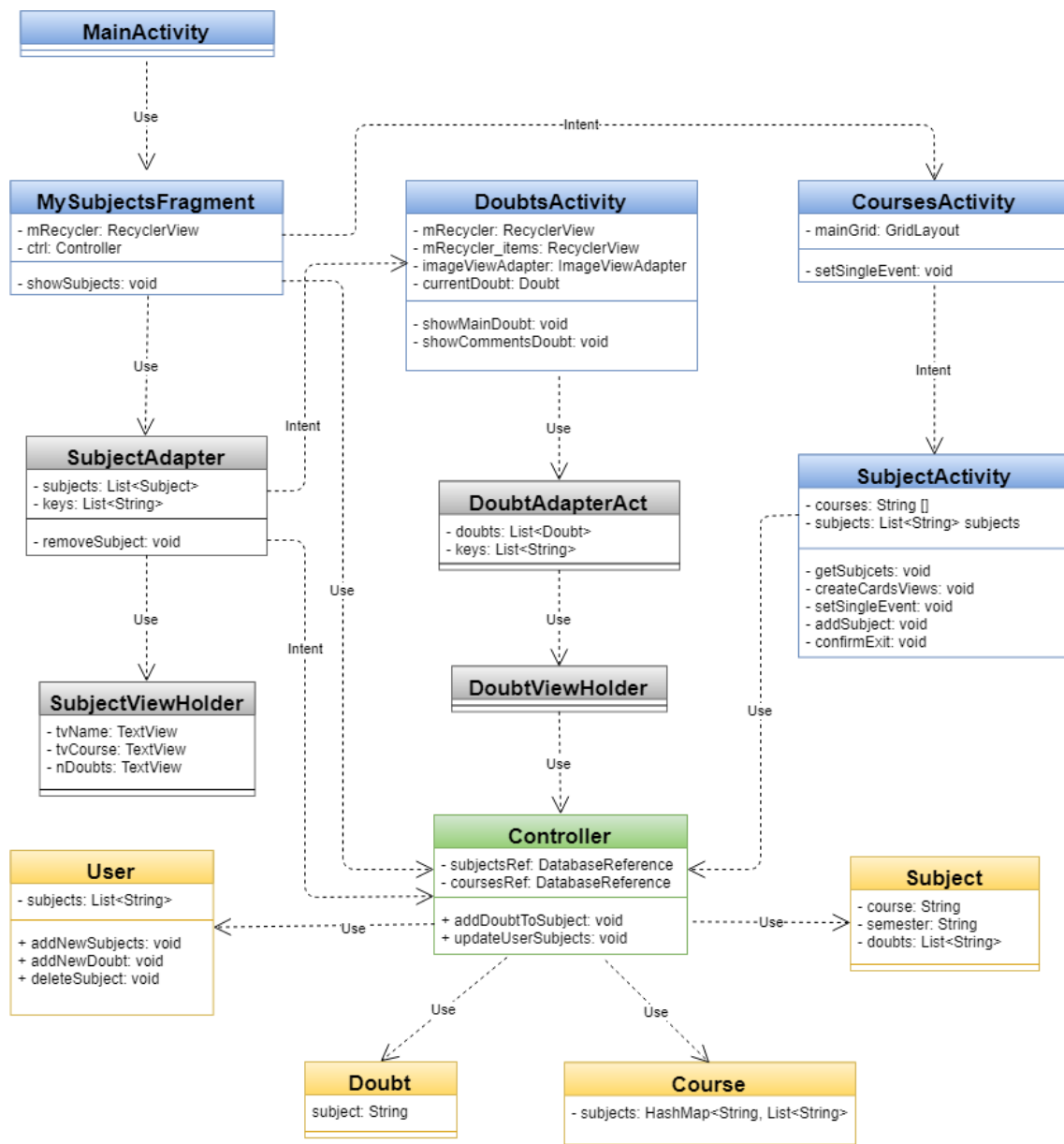


Figura 21. Diagrama de clase del incremento 3 correspondiente al modelo MVC

Siguiendo el modelo de los anteriores diagramas de clases, a continuación, vamos a comentar las clases más destacadas de este incremento:

- **MySubjectsFragments:** Este *fragment* es el único que no comparten todos los miembros de la comunidad. Cada usuario dispondrá en esta pantalla sólo las asignaturas que desea observar.
 - ShowSubjects: Este método se encarga de traer desde base de datos todas las asignaturas que contiene la lista del usuario y de mostrarlas por pantalla con el patrón ViewHolder.
- **DoubtsActivity:** Es la clase encargada de mostrar las dudas de una asignatura en concreto. Trabaja de forma similar que la clase HomeFragment pero con la diferencia de que en vez

de utilizar el `FirestoreRecyclerViewAdapter`, nos basamos en la clase `adapter` creada por nosotros mismos (`FirestoreAdapterAct`) para que así acepte una lista de nombres en vez de los objetos tal y como lo hace el de `Firestore`. Esto nos permite tener en base de datos para cada usuario una lista con los nombres de las asignaturas (Figura 22) e ir mostrando estas a partir de sus nombres y no guardas los objetos directamente. Esto lo hemos hecho para simular el funcionamiento de una base de datos relacional y trabajar de forma distinta para aprender diversas formas de gestionar los datos almacenados en `Firestore`.



Figura 22. Ejemplo de usuario en la base de datos con su lista de asignaturas

- **CoursesActivity:** Esta clase muestra los cursos que existen para así luego mostrar todas las asignaturas correspondientes a cada una de ellas.
- **SubjectActivity:** Esta clase destaca porque es la única que contiene todos los elementos que se muestran en ella creados a partir de código Java en vez de XML. Esto lo hemos hecho de esta forma para que cada vez que pulse en un curso, aparezcan las asignaturas por pantalla sin importar el número de cursos que son. Es decir, un curso puede tener sólo dos asignaturas, por lo que se generarán dos cajas para estas asignaturas en vez de ocho como puede suceder en otro curso.
 - `GetSubjects:` Obtiene todas las asignaturas del curso desde base de datos y las separa por semestre para así crear la pantalla con las asignaturas que pertenecen a un u otro semestre. Un ejemplo de tabla de los cursos y asignaturas en la base de datos se puede observar en la Figura 23.
 - `CreateCards:` Este es el método que crea las cajas de las asignaturas con su nombre y, además, deshabilita todas aquellas que el usuario ya tiene en su lista de asignaturas para así obviar estas.
 - `SetSingleEvent:` Añade a cada una de las cajas creadas la funcionalidad de ser presionadas y cambiar de color según esto.
 - `ConfirmExit:` Este método avisa al usuario de que volverá a la pantalla anterior, selección de cursos, sin haber guardado las asignaturas seleccionadas.

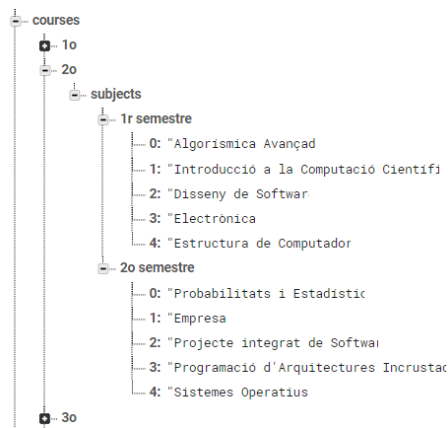


Figura 23. Ejemplo de un curso y sus asignaturas

5.3.3. EVALUACIÓN

De la misma forma que en el incremento anterior, hemos realizado una evaluación con los mismos usuarios que ya conocen la aplicación para así enseñarles el avance que ésta ha tenido y saber su opinión sobre la usabilidad y el diseño con las asignaturas.

A continuación, expondremos las características que se añadieron gracias a los entrevistados:

Las ideas que sugirieron los usuarios con los que se hicieron pruebas, son las siguientes:

- En la pantalla de seleccionar asignaturas, el color verde sustituirlo por azul, para que sea un color similar al contexto de la aplicación y también quitar el color naranja por un gris.
- Destacar el nombre de la asignatura en la caja de la duda. Poniéndolo en azul y en negrita porque en negro como estaba anteriormente se podría confundir con la descripción.
- Si no se dispone de ninguna asignatura, un mensaje en la pantalla de “Mis asignaturas” en el centro y que realice la misma función que el botón de añadir (abrir la pantalla de selección de cursos).

5.3.4. COMPORTAMIENTO

Para comenzar, al abrir la aplicación por primera vez, no se tendrá ninguna asignatura añadida por lo que la pantalla mostrará un mensaje (Figura 24A) para que el usuario presione sobre él o el botón de añadir asignaturas. Después de realizar una de las dos acciones, se abrirá la pantalla con los cursos (Figura 24B) y al seleccionar uno de estos se podrá observar todas las asignaturas que tiene ese curso (Figura 24C). Si se selecciona alguna asignatura el usuario quiere regresar, se notificará al usuario sobre que todavía no ha añadido la asignatura seleccionada (Figura 24D). Si se ha seleccionado una o más dudas y se presiona sobre el botón de confirmación, situado en la parte superior a la derecha, se enviará al usuario a la pantalla de “Mis asignaturas” con las asignaturas que había seleccionado (Figura 24E). Ya en esta pantalla

se podrá seleccionar sobre una de las asignaturas y se mostrarán las dudas que pertenecen a esta asignatura (Figura 24F) siempre y cuando el número de dudas sea mayor a cero, sino se mostrará un mensaje notificando al usuario que dicha duda no contiene ninguna duda a mostrar.

El usuario también tiene la posibilidad de borrar de la lista cualquier asignatura que desee manteniendo presionado por un momento, de la misma forma que eliminar una imagen cuando se está creando una nueva duda.

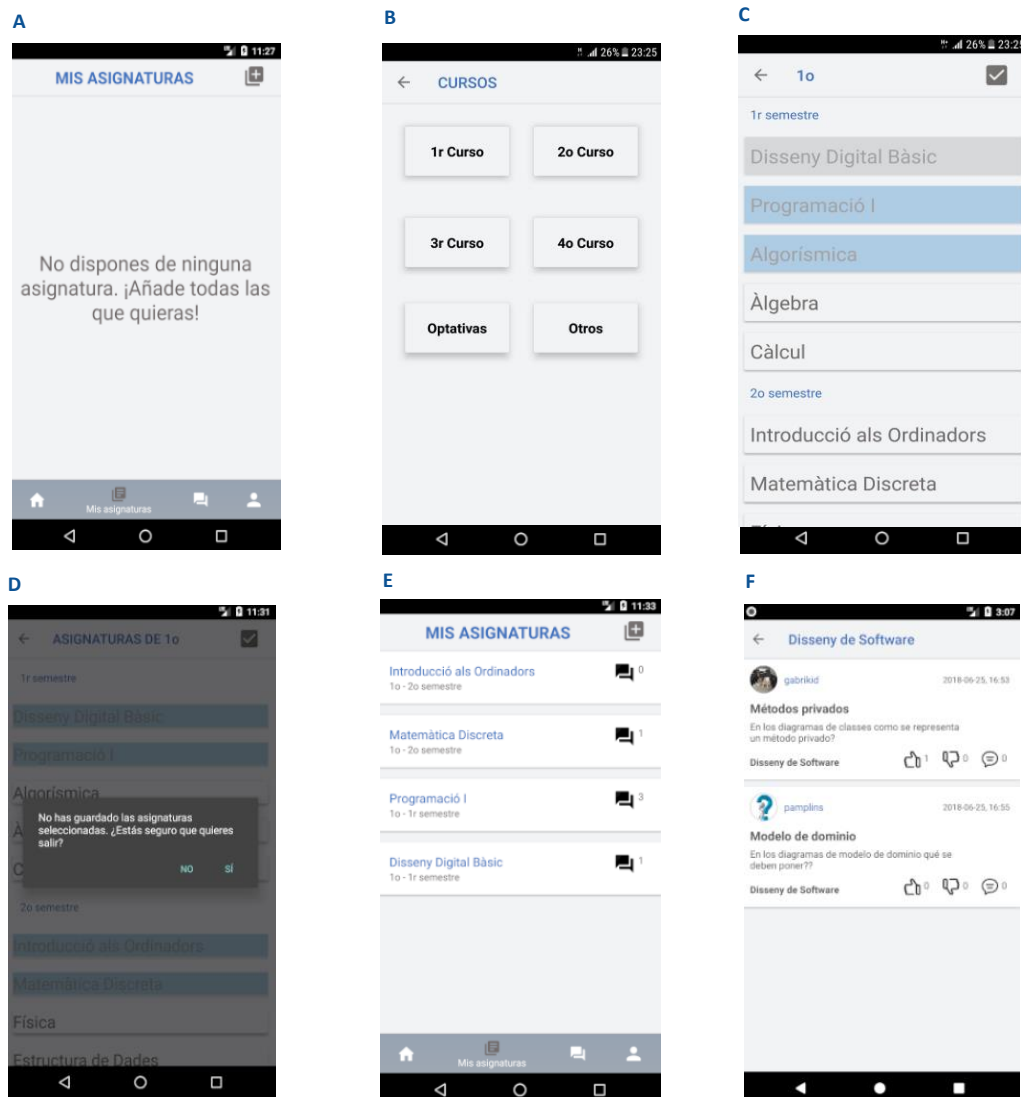


Figura 24. Pantallas destacadas del incremento 3

5.3.5. PRUEBAS DEL FUNCIONAMIENTO

- Añadir una y múltiples asignaturas.
- Ver que las dudas que se muestran en la pantalla de “Últimas asignaturas” y las de la pantalla de una asignatura en concreto se actualizan en tiempo real.
- Eliminar asignaturas comprobando la base de datos.
- Observar que diferentes usuarios tienen la pantalla de “Mis asignaturas” configurada de forma única, que no haya conflictos entre pantallas.

5.4. INCREMENTO 4: PERFIL Y RESPUESTA AVANZADA

Este último incremento tiene como objetivo principal añadir a la aplicación la parte de perfil de usuario. Con esto, conseguimos dar una parte más personal donde cada usuario pueda ser capaz de modificar la imagen, ver las dudas creadas y poder cerrar sesión. Otra característica importante de este incremento es añadir la funcionalidad de adjuntar alguna imagen a la respuesta de una duda para así dar soporte a ésta. Además, se entregará la última versión de la aplicación por lo que se hará una última evaluación con tal de saber la opinión de terceros y finalizar así la parte de desarrollo del proyecto. Por último, resaltaremos todo lo que engloba la aplicación respecto a funcionalidades.

5.4.1. REQUISITOS FUNCIONALES Y ESPECIFICACIONES

1. El usuario debe haber iniciado sesión para poder visualizar la pantalla de perfil.
2. Esta pantalla de perfil debe ser sencilla y con funcionalidades concretas. Estas funcionalidades son las siguientes:
 - Ver y modificar la foto de perfil de la misma forma que al registrarse.
 - Botón situado en la parte superior a la derecha para cerrar sesión con mensaje de confirmación.
 - Listado de todas las dudas que ha hecho el usuario para tener un control de sus propias dudas de forma fácil e instantánea.
3. La pantalla de una duda en detalle da la posibilidad de adjuntar una imagen. Esto se hará mediante una imagen con una cámara situada en la parte inferior a la izquierda, al lado de la caja de texto de la respuesta.
4. Al presionar sobre la cámara, se abre una nueva pantalla con una caja de texto más grande que da comodidad al usuario y una nueva cámara para añadir imágenes.

5.4.2. DIAGRAMA DE CLASE

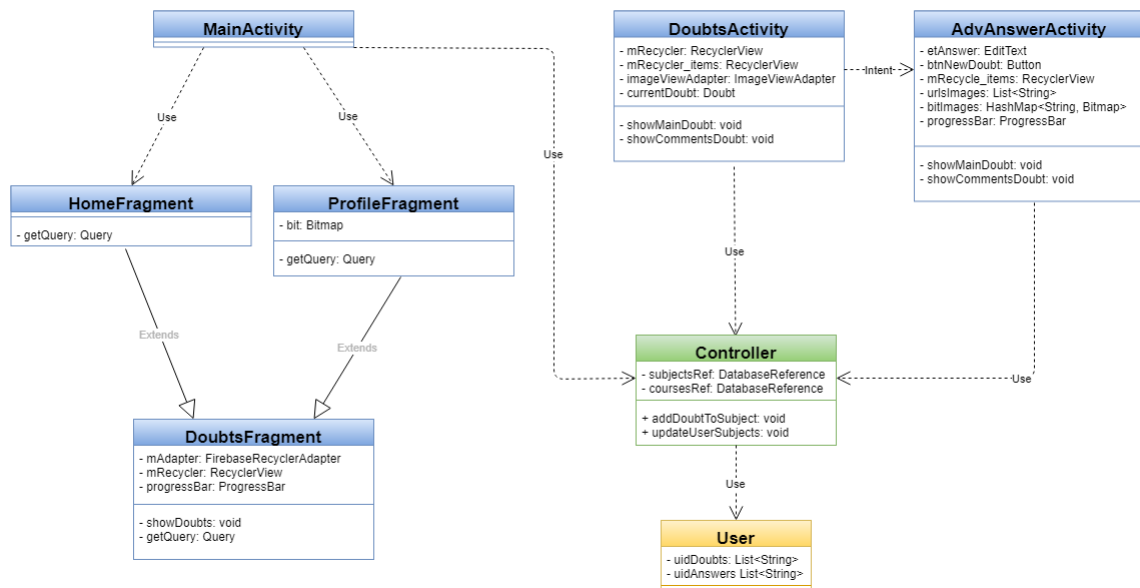


Figura 25. Diagrama de clases del incremento 4

De la misma forma que en los anteriores incrementos, a continuación, vamos a comentar las clases más destacadas de este diagrama de clases:

- **DoubtsFragment:** Esta clase se ha creado para evitar repetición de código entre las clases de HomeFragment y ProfileFragment, ya que ahora mismo ambas comparten la misma función: mostrar un listado de dudas en *fragment*, por lo que ambas clases heredan de ésta. Aunque muestren objetos dudas, la consulta a base de datos es diferente, por lo que se ha creado un método abstracto, `getQuery`, donde cada una de las clases hace su consulta determinada y comparten toda la otra información, excepto la pantalla donde se muestra el *fragment*.
- **User:** Esta clase ha tenido una adición de listas de las dudas que el usuario crea y las respuestas que da, pues ahora cada usuario puede modificar la fotografía de perfil y, por lo tanto, hay que generar una actualización de rutas múltiples, como se puede observar en la Figura 26. Debido a esto, nos guardamos cada uid generado para las dudas y respuestas, porque así facilitamos recorrer estas tablas en busca de un usuario concreto para actualizar. El método de aplicar una actualización a todas las rutas en la base de datos se denomina operación atómica, por lo que se hacen en todas las rutas o en ninguna, tal y como funciona el sistema de votaciones en las dudas y respuestas.

```

private void updateUserProfileImage(String downloadUrl) {
    Map<String, Object> updateMap = new HashMap();
    // users
    updateMap.put("/"+Constants.REF_USERS+"/"+getUid()+"/"+Constants.REF_PROFILE_NAME, downloadUrl);
    // doubts
    for(String ref: user.getUidDoubts()){
        updateMap.put("/"+Constants.REF_DOUBTS+"/"+ref+"/"+Constants.REF_PROFILE_NAME, downloadUrl);
        updateMap.put("/"+Constants.REF_USER_DOUBTS+"/"+getUid()+"/"+ref+"/"+Constants.REF_PROFILE_NAME, downloadUrl);
    }
    // answers
    String doubtRef;
    List<String> answers = user.getUidAnswers();
    for(int i = 1; i < answers.size(); i+=2){
        doubtRef = answers.get(i-1);
        updateMap.put("/"+Constants.REF_POST_ANSWERS+"/"+doubtRef+"/"+answers.get(i)+"/"+Constants.REF_PROFILE_NAME, downloadUrl);
    }
    db.getReference().updateChildren(updateMap);
}

```

Figura 26. Actualización de rutas múltiples al modificar la imagen de perfil del usuario

5.4.3. EVALUACIÓN

En esta última evaluación, hemos tenido en cuenta a los mismos usuarios que en las anteriores y cinco usuarios más. Estos nuevos usuarios tienen cierta experiencia en aplicaciones y son estudiantes de otros grados no relacionados directamente a la tecnología. Los comentarios fueron los siguientes:

- Trasladar texto de la respuesta simple a avanzada. Si un usuario empieza a escribir en la simple, quieren que automáticamente se traslade el texto a la pantalla de respuesta avanzada para así continuar y añadir una imagen si la respuesta lo requiere.
- Permitir que toda la aplicación pueda utilizarse teniendo el dispositivo móvil tanto en vertical como en horizontal.
- Una línea que separe cada respuesta.
- El color de la asignatura en la “caja” de dudas parece un *link*, por lo que modificar el color de éste.
- Al presionar un botón, que haga un efecto que demuestre que ha sido presionado.
- Añadir otros idiomas, el catalán e inglés, por ejemplo.

5.4.4. COMPORTAMIENTO

Después de iniciar la aplicación, se puede ir a la pantalla de “Mi perfil” donde se muestra la imagen que tiene el usuario, su seudónimo y las dudas que ha realizado (Figura 27A). Aquí el usuario puede modificar su foto de perfil haciendo clic sobre la imagen actual que tiene. Además, puede observar de forma rápida el estado de las dudas que ha realizado en la plataforma. Por último, esta pantalla ofrece cerrar sesión por si se desea iniciar con otra cuenta, siempre mostrando un mensaje de confirmación a la hora de realizar esta acción.

Como se puede observar en la Figura 27B, se ha añadido la funcionalidad de añadir imágenes a la respuesta, esto nos lleva a la pantalla de la Figura 27C, donde el usuario tiene un campo de texto más grande para así contestar con una mayor comodidad y puede adjuntar imágenes

que refuercen su respuesta. Una vez se ha subido la respuesta, volverá a la pantalla anterior con su respuesta añadida.

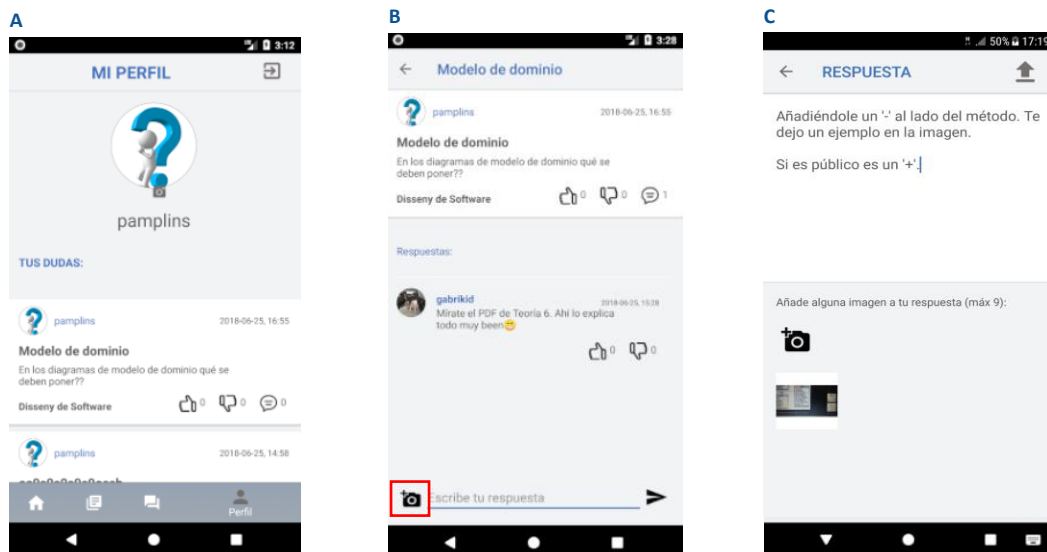


Figura 27. Pantallas destacadas del incremento 4

5.4.5. PRUEBAS DEL FUNCIONAMIENTO

1. Modificar la imagen de perfil y ver que se actualizan en todas las repuestas y dudas creadas por el usuario.
2. Cerrar sesión e ingresar de nuevo en la aplicación.
3. Guardar en una tabla de la base de datos cada una de las dudas generadas por el usuario y comprobar que la actualización de estas dudas se realiza tanto en esta tabla como en las otras que contienen información de la duda.
4. Añadir una repuesta con imágenes y comprobar el traslado de texto de la respuesta simple a avanzada para así evitar repetir lo escrito anteriormente en la simple.

Al ser el último incremento, aprovechamos a detallar las funcionalidades finales que tiene la aplicación después de finalizar el último incremento. Se ha conseguido desarrollar una aplicación que cumple con las siguientes funciones:

- Base de datos consistente, donde se almacenan las tablas con toda la información que la aplicación necesita.
- Un repositorio para guardar las imágenes subidas por toda la aplicación.
- Sistema de validación de datos en todos los casos posibles como es en registro, inicio de sesión y en la asignatura a la hora de crear una nueva duda.
- Interfaz usable, aplicando patrones de diseño propios del sistema y familiares al usuario que facilita la navegación por la aplicación.

- Aplicación funcional tanto si se tiene el dispositivo móvil en horizontal como en vertical.
- Múltiples foros donde se pueden discutir las diferentes dudas hechas por parte de los usuarios.
- Parte personal en la aplicación con las pantallas de mis asignaturas y mi perfil.

6. DISCUSIÓN

La creación de ShareDoubt surge como una herramienta de ayuda para los estudiantes de ingeniería informática de la UB por el limitado uso que estos dan al servicio que ofrece el grado para compartir alguna duda, el foro del Campus Virtual. El objetivo principal de este proyecto es desarrollar una aplicación que sustente una comunidad de estudiantes, donde puedan compartir dudas, conocimiento, ayudarse y aprender de forma conjunta. Además, en esta comunidad se premia la confianza ya que el respeto se conserva gracias al anonimato de los usuarios.

Tras varios meses de trabajo, los objetivos generales propuestos al inicio de esta memoria han podido llevarse a cabo de forma satisfactoria. No obstante, por falta de conocimiento técnicos y sobre todo por tiempo, no se ha podido llevar a cabo una petición propuesta por los usuarios entrevistados como es la de ordenar las respuestas por votación que propusieron en la indagación del prototipo. Esta característica permite a los usuarios tener la primera respuesta como referencia a la duda. El motivo de no haber podido realizar esta funcionalidad, aparte de los comentados anteriormente, es que trabajamos con un sistema de base de datos a tiempo real y cada cambio se ve afectado en el momento. Con esto queremos decir que, realmente sí logramos hacer que las respuestas estuvieran ordenadas por los votos positivos, pero, esto se hacía de forma instantánea haciendo dificultoso la posibilidad de llevar un orden de lectura, ya que las respuestas se reordenaban al momento de que hubiera una modificación en alguna de ellas. Por lo que tras opinión propia y de terceros, decidimos dejarlo a un lado.

Un dato interesante que surge al haber añadido la pantalla de “Mis asignaturas” a la aplicación en el incremento 3, es que a pesar de que sea la pantalla personal del usuario, ya que son las asignaturas que él ha escogido ver, hemos decidido que siga abriéndose como principal la pantalla de “Últimas dudas”. El objetivo de esta decisión es incentivar al usuario a ver las dudas que los compañeros han compartido sin importar que no las tengan en su lista. Con esto pensamos que generaremos más opciones de resolver las dudas de la comunidad. Si facilitamos que el usuario sólo se centre en sus asignaturas, evitará ver las que ya han cursado y seguramente pueda aportar su conocimiento en alguna duda que vea justo al abrir la aplicación.

El servidor utilizado para desarrollar la aplicación ha sido la plataforma Firebase. Ésta trabaja de forma asíncrona, de manera que los accesos a ella son lo último que se ejecuta. Es por esto por lo que el patrón MVC ha sido modificado para que se pudiera adaptar a la manera de trabajo de Firebase, convirtiéndose un MVC híbrido. Paralelamente, al trabajar con una base de datos NoSQL, tenemos una desnormalización de datos para conseguir una mayor rapidez a la hora de acceder a ellos y poder reducir consultas por su simplicidad.

Cabe comentar que el hecho de haber utilizado diferentes *adapters*, el ofrecido por Google y el creado por nosotros mismos, nos ha dado la posibilidad de observar que ambos tienen ventajas y desventajas entre sí. Lo más destacado es: el de Firebase da la posibilidad de que él sólo gestione todos los cambios que se vean afectados en la base de datos, pero no te da la opción de saber qué número de objetos hay en total. Además, te genera crear más tablas en la base de datos con objetos para poder utilizarlo. Por otro lado, el creado por nosotros te permite tener un total control de los datos que hay por tenerlos en una lista (lista de uids). Sin

embargo, hay que hacer un trabajo extra para gestionar los cambios que ocurren en la base de datos, pues tenemos que recorrer las tablas en busca de los objetos al tener sólo el uid de estos. En conclusión, queda mucho por investigar y explorar qué más nos aporta el uso de uno o el otro, pero por ahora nos decantamos por el de Firebase, ya que realiza todo de forma más fluida a pesar de tener que crear más tablas.

La metodología usada para desarrollar la aplicación, el modelo incremental, nos ha permitido ofrecer entregas de una forma práctica y robusta en cada uno de los incrementos. La evaluación de este modelo es altamente positiva, ya que ha permitido desarrollar la aplicación separándola por incrementos, de manera que siempre queda presente la evolución del trabajo. Finalmente, esto ha permitido dar una herramienta definitiva totalmente funcional.

Cabe comentar que se ha trabajado con pocos usuarios a causa del intervalo limitado de tiempo. Es por esto por lo que, para comprobar el funcionamiento a gran escala de la aplicación, se debería trabajar con un grupo de usuarios elevado y así poder evaluar las funcionalidades de la aplicación de una forma más formal y extensiva.

Por último, hay que destacar que lo aprendido en el grado nos ha sido de gran utilidad para poder llevar a cabo un proyecto de estas dimensiones, sobre todo lo comprendido en las asignaturas de programación, factores humanos y diseño de software, entre otras. Además, ha sido posible adquirir nuevos conocimientos a lo largo del desarrollo de la aplicación gracias a la base que hemos construido en el transcurso de la carrera.

7. CONCLUSIÓN

La realización de este proyecto ha sido todo un reto, ya que hemos partido con pocos conocimientos, pero finalmente hemos conseguido crear una aplicación móvil totalmente funcional que da soporte a los estudiantes a la hora de estudiar. Llegados a este punto, podemos considerar que los objetivos iniciales se han podido cumplir y las conclusiones que extraemos son las siguientes:

- La ayuda que se ofrece en las aplicaciones móviles del ámbito escolar puede llegar a ser muy efectiva y aumenta la socialización si se destina al aprendizaje colectivo y colaborativo.
- Existen diferentes aplicaciones basadas en foros, pero son de carácter general. Lo que ShareDoubt ofrece es tener en una misma comunidad a estudiantes del mismo grado donde puedan compartir sus dudas y obtener ayuda en temas específicos.
- El estudio del diseño nos ha facilitado crear una herramienta basada en elementos comunes, básicos y fáciles de usar con los que el usuario es capaz de navegar por la aplicación con total facilidad.
- La implementación de todo el *software* es la parte que, como estudiantes de ingeniería informática, más ambición y satisfacción nos ha generado. Ésta es la parte en la que se ha dedicado más tiempo durante el proyecto, lo que nos ha permitido aprender a programar una aplicación móvil Android, principal motivación de este trabajo.

Después de haber conseguido la aplicación deseada al inicio del proyecto, pensamos que todavía se pueden realizar mejoras para así entregar una herramienta que facilite aún más el estudio de los estudiantes. Estas perspectivas de futuro basadas en conseguir implementar más funcionalidades a nuestra aplicación se desarrollarán como objetivos a largo plazo.

7.1. PERSPECTIVAS DE FUTURO

El primer paso para que la aplicación sea más completa, es conseguir trasladar toda la herramienta a dispositivos con sistema operativos diferentes a Android, para así dar soporte a todos los dispositivos móviles posibles. La adición de nuevas facultades de la Universidad de Barcelona para así llegar a todos los estudiantes posibles será otro de los objetivos a cumplir. Esto ofrecería la oportunidad de tener para cada facultad la aplicación focalizada sólo en las asignaturas correspondientes a ésta.

Después de conseguir llegar a la mayoría de los dispositivos y alumnado de la universidad, añadiríamos diferentes detalles internos en la aplicación que nos llevaría a desarrollar una nueva versión de ésta. Son los siguientes:

- Administración de facultades, donde la aplicación tendría un usuario administrador en cada una de la facultad que permitiese poder añadir, modificar y eliminar las diferentes asignaturas del grado. Además, se convertiría en el moderador, vigilando que las discusiones sean siempre positivas.

- Respuesta más votada, esta funcionalidad no la hemos podido conseguir por lo que se ha transformado en una adición para que una duda pueda obtener una respuesta clave votada por la comunidad.
- Subir otros tipos de archivos, ya que actualmente contamos sólo con el soporte de imágenes para ayudar a entender las dudas y las respuestas.
- Una vez tengamos un gran volumen de datos, añadir un análisis extensivo de las asignaturas de la aplicación. De esta forma, el administrador podría observar las carencias de conocimientos en temarios o en ámbitos específicos. Esto le daría soporte al profesor de una asignatura para tomar nota de los detalles que no se están comprendiendo como le gustaría.
- Puntos de reputación para los usuarios, de esta forma cada perfil obtendría puntos según las respuestas positivas y número de dudas creadas.
- Búsqueda de foros para así evitar múltiples hilos con la misma información.

En conclusión, se ha conseguido desarrollar una aplicación totalmente funcional que cumple con todas las características básicas que se habían propuesto en un principio. De esta forma, podría ser lanzada al mercado para así llegar a un gran público y obtener el *feedback* de los usuarios con el fin de mejorar.

8. RERENCIAS

- Arias Chaves, M. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. InterSedes: Revista de las Sedes Regionales. Recuperado el 22/06/2018 de <http://www.redalyc.org/html/666/66612870011/>
- Developers Android (2018). Recuperado el 22/06/2018 de <https://developer.android.com/reference/android/support/v7/widget/RecyclerView>
- Gikas J. (2013). Mobile computing devices in higher education: Student perspectives on learning with cellphones, smartphones & social media. The Internet and Higher Education. Recuperado el 22/06/2018 de <https://www.sciencedirect.com/science/article/pii/S1096751613000262?via%3Dihub>
- Houston, H., Shewmaker, J. y Nguyen, J. (2017). Impact of mobile technology on student attitudes, engagement, and learning. Computers & Education. Recuperado el 22/06/2018 de <https://www.sciencedirect.com/science/article/pii/S0360131517300064?via%3Dihub>
- Image Popup (2017). Recuperado el 22/06/2018 de <https://github.com/chathuralakmal/AndroidImagePopup>
- Material Design (2018). Recuperado el 22/06/2018 de <https://material.io/>
- Mora, F. (2011). Foros virtuales: aspectos por considerar. Calidad en la Educación Superior. Recuperado el 22/06/2018 de <https://dialnet.unirioja.es/servlet/articulo?codigo=5580796>
- Quinzá Pérez, C. (2016). Rivence: Aplicación móvil de servicios al ciudadano. Recuperado el el 22/06/2018 de <http://hdl.handle.net/10251/68998>
- Sommerville, I. (2011). Software process models. Software engineering-Pearson. Ingeniería del Software. Recuperado el 22/06/2018 de https://edisciplinas.usp.br/pluginfile.php/2150022/mod_resource/content/1/1429431793.203Software%20Engineering%20by%20Somerville.pdf
- Y.-J. Song, S.-b. Ou and J.-w. Lee (2016) "An Analysis of Existing Androdid Image Loading Libraries: Picasso, Glide, Fresco, AUIL and Volley," in International Conference on Informatics, Management Engineering and Industrial Application (IMEIA 2016)

9. ANEXOS

9.1. ENCUESTA GOOGLE FORMS

FORUM APP

Formulario para saber la opinión del alumnado y sobre lo qué opinas de tener una aplicación donde se puedan preguntar dudas y resolver estas, a través de un foro, sobre las diferentes asignaturas de las carreras de la Universidad de Barcelona.

Este foro será a tiempo real y podrás compartir información mediante texto, imágenes o documentos. Habrán diferentes foros para cada una de las diferentes asignaturas y así tener organizadas todas las cuestiones que surjan.

Debes saber, que por el hecho de contestar este cuestionario (anónimo y de investigación universitaria) nos proporcionas el permiso de utilizar las respuestas como fuente de información, imágenes del cuestionario respondido y finalmente, conclusiones para mejorar una futura interfaz enfocada a los usuarios.

***Obligatorio**

1. ¿Eres de los que tienen vergüenza preguntar dudas en clase? *

Marca solo un óvalo.

- ☐ Sí
☐ No

2. ¿Te gustaría poder tener la opinión de compañeros que hayan hecho o estén cursando la asignatura? *

Marca solo un óvalo.

- ☐ Sí
☐ No

3. ¿Qué herramientas sueles utilizar para compartir tus dudas? *

Marca solo un óvalo.

- ☐ Campus Virtual
☐ Whatssap
☐ Facebook
☐ Otros

4. ¿Te importaría compartir apuntes si es de forma recíproca? *

Marca solo un óvalo.

- ☐ Sí
☐ No

5. ¿Sueles compartir tu duda con un compañero o un profesor? *

Marca solo un óvalo.

- ☐ Profesor
☐ Alumno

6. ¿Te importaría compartir apuntes si es de forma recíproca? *

Marca solo un óvalo.

- ☐ Sí
☐ No

7. ¿Con qué frecuencia sueles utilizar el foro del Campus Virtual? *

Marca solo un óvalo.

- ☐ Alta
☐ Normal
☐ Baja

8. ¿Hacia quién crees que está destinado el foro del Campus Virtual? *

Marca solo un óvalo.

- ☐ Profesor
☐ Alumno

9. ¿A qué crees que es debido el poco uso del foro del Campus Virtual? *

Marca solo un óvalo.

- ☐ Timidez
☐ Preguntar a profesor
☐ Poco activo

10. ¿Utilizarías una app que te permita compartir tus dudas de forma anónima dentro comunidad de alumnos? *

Marca solo un óvalo.

- ☐ Sí
☐ No